

Министерство образования  
Российской Федерации

Оренбургский государственный университет

*В.И. Чепасов,  
Д.Р. Мустафина, Д.А. Харченко*

**Алгоритмическая и программная  
реализация метода наименьших квадратов  
в длинной арифметике**

**Оренбург-2009**

ББК 30.121: 22.172  
Ч- 44  
УДК 539.3/.6.001.24:519.2

Рецензент:  
заслуженный деятель науки, д.т.н., профессор Шевеленко В.Д.

**Ч- 44 Чепасов В.И.**

**Алгоритмическая и программная реализация метода  
наименьших квадратов в длинной арифметике – Оренбург:  
ОГУ, 2009. - 351 с.**

Ч 2402000000

ББК 30.121: 22.172

© Чепасов В.И., 2009  
ОГУ, 2009

## СОДЕРЖАНИЕ

	Стр.
<b>ВВЕДЕНИЕ</b>	<b>5</b>
<b>Глава-1 Реализация метода наименьших квадратов в длинной арифметике</b>	<b>6</b>
<b>1.1 Операция сложения(язык Фортран)</b>	<b>6</b>
<b>1.2 Операция вычитания(язык Фортран)</b>	<b>8</b>
<b>1.3 Операция умножения(язык Фортран)</b>	<b>9</b>
<b>1.4 Операция деления(язык Фортран)</b>	<b>11</b>
<b>1.5 Программная реализация метода наименьших квадратов (язык Фортран)</b>	<b>12</b>
<b>1.6 Входные и выходные данные программы(язык Фортран)</b>	<b>14</b>
<b>1.7 Поставка программного комплекса(язык Фортран)</b>	<b>15</b>
<b>1.8 Программная реализация метода наименьших квадратов в «С++»</b>	<b>17</b>
<b>1.9 Входные и выходные данные программы(язык С++)</b>	<b>21</b>
<b>1.10 Поставка программного комплекса</b>	<b>25</b>
<b>Глава 2</b>	<b>26</b>
<b>Алгоритмическое описание программного комплекса (язык Фортран)</b>	
<b>2.1 Подпрограмма по реализации сложения без учета знаков слагаемых</b>	<b>26</b>
<b>2.2 Подпрограмма по реализации сложения с учетом знаков слагаемых</b>	<b>28</b>
<b>2.3 Подпрограмма по реализации вычитания без учета знаков уменьшаемого и вычитаемого</b>	<b>29</b>
<b>2.4 Подпрограмма по реализации вычитания</b>	

с учетом знаков уменьшаемого и вычитаемого	32
<b>2.5 Подпрограмма по реализации умножения без учета знаков сомножителей</b>	<b>33</b>
<b>2.6 Подпрограмма умножения с учетом знаков сомножителей</b>	<b>35</b>
<b>2.7 Подпрограмма по реализации деления без учета знаков делимого и делителя</b>	<b>36</b>
<b>2.8 Подпрограмма по реализации деления с учетом знаков делимого и делителя</b>	<b>38</b>
<b>2.9 Подпрограмма по реализации прямого хода в методе Гаусса</b>	<b>39</b>
<b>2.10 Подпрограмма по нахождению главного элемента</b>	<b>41</b>
<b>2.11 Подпрограмма по реализации обратного хода в методе Гаусса</b>	<b>43</b>
<b>2.12 Подпрограмма по проверке решения системы методом Гаусса</b>	<b>45</b>
<b>2.13 Подпрограмма <math>m \times n \times k</math></b>	<b>46</b>
<b>2.14 Главная программа <math>d \times l \times m \times n \times k \times 2</math></b>	<b>46</b>
<b>Список литературы</b>	<b>47</b>
<b>ПРИЛОЖЕНИЕ-1(наборы с исходными текстами программ на Фортране, исходные и выходные данные для программы на Фортране)</b>	<b>48</b>
<b>ПРИЛОЖЕНИЕ-2(наборы с исходными текстами программ на C++, исходные и выходные данные для программы на C++)</b>	<b>144</b>
<b>ПРИЛОЖЕНИЕ-3(программа по реализации метода наименьших квадратов в длинной арифметике с распараллеливанием прямого хода в методе Гаусса,исходные и выходные данные)</b>	<b>249</b>

## Введение

На сегодняшний день все современные ЭВМ имеют конечную разрядную сетку.

В связи с этим невозможно осуществлять вычисления с необходимой степенью точности.

Поэтому был разработан алгоритм и осуществлена его программная реализация для выполнения всех арифметических операций с произвольным количеством разрядов в целой и дробной частях операндов.

На базе длинной арифметики был реализован метод наименьших квадратов для построения многомерных моделей аппроксимации.

Программная реализация была сделана на алгоритмических языках Фортран, С++ .

## **Глава-1 Реализация метода наименьших квадратов в длинной арифметике**

Для реализации метода наименьших квадратов в длинной арифметике все арифметические операции сложение, вычитание, деление, умножение были реализованы в длинной арифметике для вещественных значений.

Для этого были организованы для каждого числа два массива: массив, каждый элемент которого содержал разряд целой части числа, массив, каждый элемент которого содержал разряд дробной части числа.

Кроме этого для каждого числа вводилась целая переменная, которая содержала знак этого числа.

Значение 1 соответствует положительному значению, значение (-1) отрицательному значению.

### **1.1 Операция сложения(язык Фортран)**

Непосредственное сложение двух вещественных значений осуществляет подпрограмма `sl.for` (язык Фортран, приложение-1).

Сложение двух вещественных значений с учетом знаков осуществляет подпрограмма `slzn.for`.

В подпрограмме `slzn` осуществляется вызов подпрограммы `sl.for` для непосредственного сложения.

Вызов подпрограммы `slzn.for`:

```
call slzn (iz1, iz2, izrez, nc1, nd1, ic1, id1, nc2, nd2, ic2, id2, kcrez, kdrez, icrez, idrez, krazr)
```

Параметры при вызове подпрограммы `slzn`:

`iz1` – знак первого слагаемого, целая четырехбайтовая переменная;

`iz2` – знак второго слагаемого, целая четырехбайтовая переменная;

`izrez` – знак вычисленной суммы, целая четырехбайтовая переменная;

nc1 – количество разрядов в целой части первого слагаемого, целая четырехбайтовая переменная;

nd1 – количество разрядов в дробной части первого слагаемого, целая четырехбайтовая переменная;

ic1 – массив значений разрядов в целой части первого слагаемого, целые двухбайтовые;

id1 – массив значений разрядов в дробной части первого слагаемого, целые двухбайтовые;

nc2 – количество разрядов в целой части второго слагаемого, целая четырехбайтовая;

nd2 - количество разрядов в дробной части второго слагаемого, целая четырехбайтовая;

ic2 - массив значений разрядов в целой части второго слагаемого, целые двухбайтовые;

id2 - массив значений разрядов в дробной части второго слагаемого, целые двухбайтовые;

kcrez – количество разрядов в целой части суммы, целое четырехбайтовое;

kdrez - количество разрядов в дробной части суммы, целое четырехбайтовое;

icrez - массив значений разрядов в целой части суммы, целые двухбайтовые;

idrez - массив значений разрядов в дробной части суммы, целые двухбайтовые;

krazi- количество разрядов в дробной части длинной арифметики, целое четырехбайтовое.

Пользователю для реализации длинного сложения надо вызвать подпрограмму slzn.

## 1.2 Операция вычитания(язык Фортран)

Вычитание двух вещественных с учетом знаков осуществляет подпрограмма `wiczn.for`(приложение-1).

В подпрограмме `wiczn` осуществляется вызов подпрограммы `wich` для непосредственного вычитания.

Вызов подпрограммы `wiczn`:

**call wiczn (iz1, iz2, izrez, nc1, nd1, ic1, id1, nc2, nd2, ic2, id2, kcrez, kdrez, icrez, idrez,krazr),**

`iz1` - знак уменьшаемого, целая четырехбайтовая,

`iz2` – знак вычитаемого, целая четырехбайтовая,

`izrez` – знак разности, целая четырехбайтовая,

`nc1` – количество разрядов в целой части уменьшаемого, целая четырехбайтовая;

`nd1` – количество разрядов в дробной части уменьшаемого, целая четырехбайтовая;

`ic1` – массив значений разрядов в целой части уменьшаемого, целые двухбайтовые;

`id1` - массив значений разрядов в дробной части уменьшаемого, целые двухбайтовые;

`nc2` – количество разрядов в целой части вычитаемого, целое четырехбайтовое;

`nd2` - количество разрядов в дробной части вычитаемого, целое четырехбайтовое;

`ic2` – массив значений разрядов в целой части вычитаемого, целые двухбайтовые;

id2 - массив значений разрядов в дробной части вычитаемого, целые двухбайтовые;

kcrez - количество разрядов в целой части разности, целое четырехбайтовое;

kdrez - количество разрядов в дробной части разности, целое четырехбайтовое;

icrez - массив значений разрядов в целой части разности, целые двухбайтовые;

idrez - массив значений разрядов в дробной части разности, целые двухбайтовые;

krazr-количество разрядов в дробной части длинной арифметики, целое четырехбайтовое.

Для реализации длинного вычитания пользователю необходимо вызвать подпрограмму wiczp.

### **1.3 Операция умножения(язык Фортран)**

Умножение двух вещественных в длинном формате осуществляет подпрограмм umnzn.for (с учетом знаков)(приложение-1).

В подпрограмме umnzn осуществляется вызов подпрограммы umn для непосредственного умножения в длинном формате.

Вызов подпрограммы umnzn:

**call umnzn (iz1, iz2, izrez, nc1, nd1, ic1, id1, nc2, nd2, ic2, id2, kcrez, kdrez, icrez, idrez,krazr),**

iz1 - знак первого сомножителя, целое четырехбайтовое;

iz2 – знак второго сомножителя, целое четырехбайтовое;

izrez – знак произведения, целое четырехбайтовое;

nc1 – количество разрядов в целой части первого сомножителя, целое четырехбайтовое;

nd1 – количество разрядов в дробной части первого сомножителя, целое четырехбайтовое;

ic1 – массив значений разрядов в целой части первого сомножителя, целые двухбайтовые;

id1 - массив значений разрядов в дробной части первого сомножителя, целые двухбайтовые;

nc2 – количество разрядов в целой части второго сомножителя, целое четырехбайтовое;

nd2 - количество разрядов в дробной части второго сомножителя, целое четырехбайтовое;

ic2 – массив значений разрядов в целой части второго сомножителя, целые двухбайтовые;

id2 - массив значений разрядов в дробной части второго сомножителя, целые двухбайтовые;

kcrez - количество разрядов в целой части произведения, целое четырехбайтовое;

kdrez - количество разрядов в дробной части произведения, целое четырехбайтовое;

icrez - массив значений разрядов в целой части произведения, целые двухбайтовые;

idrez - массив значений разрядов в дробной части произведения, целые двухбайтовые;

krazi-количество разрядов в дробной части длинной арифметики, целое четырехбайтовое.

Для реализации длинного умножения пользователю надо вызвать подпрограмму umnzn.

## 1.4 Операция деления(язык Фортран)

Длинное деление осуществляет подпрограмма `delzn.for`(приложение-1) с учетом знаков вещественных значений.

Непосредственное деление осуществляет подпрограмма `deld`, которая вызывается в подпрограмме `delzn`.

Вызов подпрограммы `delzn`:

```
call delzn (iz1, iz2, izrez, nc1, nd1, ic1, id1, nc2, nd2, ic2, id2, kcrez, kdrez, icrez, idrez,krazr),
```

`iz1` - знак делимого, целое четырехбайтовое;

`iz2` – знак делителя, целое четырехбайтовое;

`izrez` – знак частного, целое четырехбайтовое;

`nc1` – количество разрядов в целой части делимого, целое четырехбайтовое;

`nd1` – количество разрядов в дробной части делимого, целое четырехбайтовое;

`ic1` – массив значений разрядов в целой части делимого, целые двухбайтовые;

`id1` - массив значений разрядов в дробной части делимого, целые двухбайтовые;

`nc2` – количество разрядов в целой части делителя, целое четырехбайтовое;

`nd2` - количество разрядов в дробной части делителя, целое четырехбайтовое;

`ic2` – массив значений разрядов в целой части делителя, целые двухбайтовые;

id2 - массив значений разрядов в дробной части делителя, целые двухбайтовые;

kcrez - количество разрядов в целой части частного, целое четырехбайтовое;

kdrez - количество разрядов в дробной части частного, целое четырехбайтовое;

icrez - массив значений разрядов в целой части частного, целые двухбайтовые;

idrez - массив значений разрядов в дробной части частного, целые двухбайтовые;

krazg-количество разрядов в дробной части длинной арифметики, целое четырехбайтовое.

Для реализации длинного деления пользователю надо вызвать подпрограмму delzn.

### **1.5 Программная реализация метода наименьших квадратов(язык Фортран)**

С помощью подпрограмм gaydl.for, nimax.for, obrdl.for, prowg.for, slzn, wiczsl, wich, ymnzn, ymn, delzn, deld реализован метод наименьших квадратов в длинной арифметике.(приложение-1).

В главной программе dlmnk2.for осуществляется ввод матрицы исследования, названий параметров-столбиков матрицы исследования, номеров параметров-аргументов, номеров зависимых параметров, степени аппроксимирующего полинома, количества разрядов в дробной части длинной арифметики, вызов подпрограммы mnk.

В подпрограмме mnk осуществляется построение матрицы системы нормальных уравнений, правых частей этой системы, вызов полпрограммы gaydl.for

для реализации метода Гаусса, вызов подпрограммы `prowg.for` для проверки решения системы.

В подпрограмме `gaydl.for` осуществляется вызов двух подпрограмм `nimax.for`, `obrdl.for`.

В подпрограмме `gaydl.for` идет реализация прямого хода метода Гаусса, в подпрограмме `nimax.for` нахождение главного элемента, в подпрограмме `obrdl.for` - обратный ход метода Гаусса.

Вызов подпрограммы `gaydl`:

**call gaydl (n, znak, nc, nd, ac, ad, znr, nbc, nbd, bc, bd, znx, nxc, nxd, xc, xd, igau, krazr ),**

`n` – порядок матрицы системы;

`znak (i, j)` – знак элемента  $c_{ij}$  в матрице системы, целое двухбайтовое значение;

`nc (i, j)` – количество разрядов в целой части элемента  $c_{ij}$  в матрице системы, целое двухбайтовое;

`nd (i, j)` – количество разрядов в дробной части элемента  $c_{ij}$  в матрице системы, целое двухбайтовое;

`ac (i, j, l)` – массив значений разрядов в целой части элемента  $c_{ij}$  матрицы системы, целые двухбайтовые;

`ad (i, j, l)` – массив значений разрядов дробной части  $c_{ij}$  матрицы системы, целые двухбайтовые;

`znr (i)` – массив значений знаков правых частей системы  $b_i$ , целые двухбайтовые;

`nbc (i)` – массив длин целой части правых частей системы  $b_i$ , целые двухбайтовые;

`nbd (i)` – массив длин дробной части правых частей системы  $b_i$ , целые двухбайтовые;

$bc(i, l)$  – массив значений разрядов целой части свободного члена  $b_i$ , целые двухбайтовые;

$bd(i, l)$  – массив значений разрядов дробной части свободного члена  $b_i$ , целые двухбайтовые;

$zpx(i)$  – массив знаков неизвестных системы  $x_i$ , целые двухбайтовые;

$pxc(i)$  – массив длин целой части неизвестных  $x_i$ , целые двухбайтовые;

$pxd(i)$  – массив длин дробной части неизвестных  $x_i$ , целые двухбайтовые;

$xc(i, l)$  – массив значений разрядов целой части неизвестных  $x_i$ , целые двухбайтовые;

$xd(i, l)$  – массив значений разрядов дробной части неизвестных  $x_i$ , целые двухбайтовые;

$igau$  - индикатор ошибки при решении системы. Если  $ugau = 0$ , то система вырождена, если  $ugau = 1$ , то система имеет единственное решение.  $igau$  - целое четырехбайтовое;

$krazg$ -количество разрядов в дробной части длинной арифметики, целое четырехбайтовое.

## **1.6 Входные и выходные данные программ(язык Фортран)**

В главной программе `dlnk2.for` происходит чтение исходных данных из входного файла `sled` и входного файла `name`.

Файл `sled` имеет следующую структуру(приложение-1):

- в первой строчке-записи идет название матрицы исследования(до 80-ти знаков);

- во второй строчке через пробел идут количество строчек-наблюдений и количество параметров-столбиков в матрице исследования;

- в следующей строчке-записи идут через пробел элементы первой строки матрицы исследования;

-в следующей строчке-записи идут через пробел элементы второй строки матрицы исследования и т.д. элементы всех строк матрицы исследования.

В файле name в первой строчке-записи идет количество параметров-столбиков в матрице исследования(приложение-1).

Во второй строчке-записи идет название первого параметра-столбика в матрице исследования(не более 60-ти знаков).

В третьей строчке-записи идет название второго параметра-столбика в матрице исследования и т.д. в последующих строчках названия всех параметров-столбиков матрицы исследования.

Решение системы нормальных уравнений в методе наименьших квадратов и ее проверка записываются в файл growfdl2(приложение-1).

Сама модель аппроксимации и вклады параметра-аргументов записываются в файл dlmnk2(приложение-1).

## **1.7 Поставка программного комплекса(язык Фортран)**

В поставку входят исходные модули всех подпрограмм и исходный модуль главной программы:

delzn.for – длинное деление с учетом знака. В delzn идет подпрограмма deld по непосредственному делению.

slzn.for – длинное сложение с учетом знака.

sl.for – непосредственное длинное сложение.

wiczn.for – длинное вычитание с учетом знака.

wich.for – непосредственное длинное вычитание.

ymnzn.for – длинное умножение с учетом знака. В ymnzn идет подпрограмма ymn по непосредственному длинному умножению.

gaydl.for – подпрограмма по реализации прямого хода метода Гаусса в длинной арифметике. В подпрограмме gaydl идет вызов подпрограмм nimax, obrdl.

nimax.for – подпрограмма по нахождению главного элемента в длинной арифметике.

obrdl.for – подпрограмма по реализации обратного хода метода Гаусса в длинной арифметике.

mnk.for- подпрограмма, в которой осуществляется построение матрицы и свободных членов системы нормальных уравнений, вызов подпрограммы gaydl.for и подпрограммы prowg.for по проверке решения системы.

dlnmk2.for – главная программа для реализации метода наименьших квадратов в длинной арифметике. В ней осуществляется вызов подпрограммы mnk.for.

Вместе с исходными текстами программ идет поставка библиотеки под названием bibl2 и ее оглавления под названием obibl2 с объектными модулями программ.

- delzn.obj,
- slzn.obj,
- sl.obj,
- wiczn.obj,
- wich.obj,
- ymnzn.obj,
- gaydl.obj,
- nimax.obj,
- obrdl.obj,
- prowg.obj,

В поставку входит архив dlnmkfor.arj, в котором есть еще загрузочный модуль dlnmk2.exe.

Кроме этого в поставку входят файлы sled,name с данными контрольного примера.

**Замечание.**

Поставляемый комплекс позволяет реализовать в длинной арифметике метод наименьших квадратов для матриц исследования порядка 50x20(50 строчек,20 столбиков). Максимальное количество разрядов в целой и дробной частях – 20.

Максимальное количество параметров-аргументов -10,максимальный порядок аппроксимирующих полиномов-3.

**1.8 Программная реализация метода наименьших квадратов в «С++»**

В случае реализации в «С++» имеем следующие внешние вызываемые функции(приложение-2):

`mnk,gaydl,nimax,obrdl,prowg,delzn,deld,sl,slzn,wich,wiczn,ymnzn, ymn.`

Главная функция и оконная функция в наборе `dlmnkbc.cpp`.

В наборе `dlmnkc.cpp` находится внешняя вызываемая функция,в которой осуществляется ввод исходных данных,матрицы исследования,вызов `mnk`.

Функции `mnk,gaydl,nimax,obrdl,prowg, delzn,deld, sl,slzn, wich, wiczn, ymnzn, ymn` полностью аналогичны подпрограммам под теми же именами в Фортране .

Но к списку параметров внешних вызываемых функций `mnk, gaydl, nimax,obrdl,prowg` добавляется параметр `k50`,который определяет количество разрядов в целой и дробной частях при динамическом выделении памяти, и массив номеров неизвестных `mpreg` для внешней вызываемой функции `gaydl`.

В приложении-3 приводятся модификации внешних вызываемых функций для реализации метода наименьших квадратов в длинной арифметике с параллеливанием прямого хода в методе Гаусса.

Список параметров внешних вызываемых функций `delzn,deld,sl, slzn, wich, wiczn,ymnzn,ymn` полностью аналогичен списку параметров соответствующих подпрограмм в Фортране.

Но передача выходных параметров знак результата арифметической операции, количество разрядов в целой части результата, количество разрядов в дробной части результата арифметической операции идет по указателям.

Все параметры-целые двухбайтовые переменные.

Приведем прототипы внешних вызываемых функций `gaydl`, `nimax`, `obrdl`, `prowg`:

```
void gaydl(int n,int *znak,
int *nc,int *nd,
int *ac,int *ad,
int *znp,int *nbc,int *nbd,
int *bc,int *bd,
int *znx,int *nxc,int *nxd,
int *xc,int *xd,
int *pigau,int krazr,int k50,int *mnper);
```

```
void nimax(int i,int j,int n,
int *nc ,int *nd , int *znak ,
int *ac ,int *ad ,
int *nbc ,int *nbd , int *znp ,
int *bc ,int *bd ,
int *mnper, int krazr,int k50);
```

```
void obrdl(int n,int *znak ,
int *nc ,int *nd ,
int *ac ,int *ad ,
int *znp ,int *nbc ,int *nbd ,
int *bc ,int *bd ,
```

```
int *znx ,int *nxc ,int *nxd ,
int *xc ,int *xd ,
int *pigau,int *mnper,int anc[400],
int andr[400],int krazr,int k50);
```

```
void prowg(int n,int *znak ,
int *nc ,int *nd ,
int *ac ,int *ad ,
int *znp ,int *nbc ,int *nbd ,
int *bc ,int *bd ,
int *znx ,int *nxc ,int *nxd ,
int *xc ,int *xd ,
int *pigau,int krazr,int k50);
```

Во внешних вызываемых функциях `gaydl`, `obrdl`, `prowg`:  
`rigau`-указатель индикатора ошибки при решении системы линейных алгебраических уравнений.

Массивы `anc`, `andr`-это соответственно целая и дробная часть нулевого значения.

Прототипы внешних вызываемых функций по реализации длинной арифметики `delzn`, `deld`, `slzn`, `sl`, `wiczn`, `wich`, `ymnzn`, `ymn`:

```
void delzn(int za,int zb,int *pzc,
int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr);
```

```
void deld(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr);
```

```
void slzn(int za,int zb,int *pzc,  
int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);
```

```
void sl(int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);
```

```
void wiczn(int za,int zb,int *pzc,  
int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);
```

```
void wich(int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);
```

```
void ymnzn(int za,int zb,int *pzc,int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr);
```

```
void ymn(int n,int nd,int *a,int *ad,  
int m,int md,int *b,int *bd,  
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr);
```

Во внешних вызываемых функциях delzn,deld:

prc-указатель для знака частного,

pkcdel-указатель для количества разрядов в целой части частного,

pkddel-указатель для количества разрядов в дробной части частного.

Во внешних вызываемых функциях slzn, sl, wiczn, wicn, ymnzn, ymn:

prc-указатель для знака результата суммирования, вычитания, умножения,  
pkcel-указатель для количества разрядов в целой части результата суммирования, вычитания, умножения,

pkdro-указатель для количества разрядов в дробной части результата суммирования, вычитания, умножения.

Остальные параметры во внешних вызываемых функциях полностью аналогичны параметрам в соответствующих подпрограммах языка Фортран.

Во внешней вызываемой функции dlmnkbc(набор dlmnkc.cpp) идет динамическое распределение памяти для массивов(все приведены к одномерным в программной реализации).

Элементы всех массивов, все переменные-целые двухбайтовые.

Получение загрузочного модуля идет через проект.

Проект составлялся из исходных модулей внешних вызываемых функций.

В приложении-3 приводится текст программы по реализации метода наименьших квадратов в длинной арифметике с распараллеливанием прямого хода метода Гаусса, выходные результаты(до 20-ти разрядов в целой и дробной части).

### **1.9 Входные и выходные данные программы(язык C++)**

Для функционирования программы по реализации метода наименьших квадратов в длинной арифметике в среде BORLAND C++ необходимо в текстовом редакторе создать файл sleddlс следующей структуры(приложение-2):

-в первой строчке-записи идет название матрицы исследования(длина 80 символов и ENTER),

-во второй строчке-записи через пробел идут количество строчек и количество столбиков в матрице исследования(количество строчек не более 100, количество столбиков не более 100),

-в третьей строчке идет количество параметров-аргументов(не более 20),

-в четвертой строчке идут через пробел номера параметров-аргументов,

-в пятой строчке идут через пробел начальный и конечный номер зависимой переменной(если зависимая переменная одна, то начальный и конечный номер будут одинаковые),

-в шестой строчке идет степень аппроксимирующего полинома(не более 4),

-в седьмой строчке идут через пробел количество разрядов в дробной части для длинной арифметики и количество разрядов в целой и дробной частях при выделении памяти.

Количество разрядов в дробной части для длинной арифметики не должно превышать количество разрядов в целой и дробной частях при выделении памяти.

Для матрицы исследования  $100 \times 100$  (100 строчек, 100 столбцов) количество разрядов в дробной части для длинной арифметики и количество разрядов в целой и дробной частях при выделении памяти не должно превышать 50.

При этом максимальное количество разрядов в целой и дробной частях при выделении памяти-50.

-в восьмой и последующих строчках идут через пробел элементы первой строки матрицы исследования, потом элементы второй строки матрицы исследования и т.д.

Кроме файла `sleddlc` необходимо в текстовом редакторе создать файл `name`.

Файл `name` имеет следующую структуру:

-в первой строчке идет количество параметров-столбиков в матрице исследования,

-во второй строчке идет название первого параметра-столбика в матрице исследования(60 символов и ENTER),

-в третьей строчке идет название второго параметра-столбика в матрице исследования и т.д. идут построчно названия всех параметров-столбиков матрицы исследования(приложение-2).

Результат работы программы по реализации метода наименьших квадратов в длинной арифметике будет в файле dlmnkbc,который смотрим в любом текстовом редакторе(приложение-2).

В файле growcdl2 будут результаты проверки решения систем методом Гаусса(аналог файла growfdl2).

Имя загрузочного модуля по реализации метода наименьших квадратов в длинной арифметике в BORLAND C++ dlmnkbc.exe,создание которого идет через проект исходных модулей всех внешних вызываемых функций.

Для создания меню используется файл ресурсов dlmnkbc.rc и файл dlmnkbc.h с определениями номеров пунктов меню.

Входные данные для метода наименьших квадратов готовились в обычном представлении вещественных значений.

Разряды целой и дробной частей вещественных значений выделялись непосредственно в программах после ввода этих значений.

Процедура выделения разрядов дробной части может идти не совсем точно из-за специфики внутреннего представления вещественных значений в ЭВМ и из-за языковых специфик.

В связи с этим была сделана реализация поразрядного ввода целой и дробной частей вещественных значений.Это дало возможность исключить все неточности,связанные с процедурой выделения разрядов из вещественных значений.

Реализацию поразрядного ввода в фортране осуществляет программа dlmnfraz.exe(исходный модуль dlmnfraz.for)(приложение-1),в языке C++-программа dlmncraz.exe(исходный модуль dlmncraz.cpp)(приложение-2).

Для функционирования этих программ необходимо в текстовом редакторе подготовить два файла name,sledraz.

Файл name содержит названия параметров-столбиков матрицы исследования и был рассмотрен выше.

Файл sledraz содержит матрицу исследования и имеет следующую структуру:

- в первой строчке идет название матрицы исследования(80 символов и ENTER),
- во второй строчке через пробел идут количество строчек и количество столбиков в матрице исследования,
- в третьей строчке идет количество параметров-аргументов(для программы dlmnfraz не более 10,для программы dlmncraz не более 20),
- в четвертой строчке через пробел идут номера параметров-аргументов,
- в пятой строчке идут через пробел начальный и конечный номер зависимой переменной(если зависимая переменная одна,то начальный и конечный номер совпадают),
- в шестой строчке идет степень аппроксимирующего полинома(для программы dlmnfraz не более 3,для программы dlmncraz не более 4),
- в седьмой строчке идут через пробел количество разрядов в дробной части длинной арифметики и количество разрядов при выделении памяти(смотри выше) для программы dlmncraz и количество разрядов в дробной части длинной арифметики и через пробел такое же значение для программы dlmnfraz,
- в восьмой строчке идут через пробел знак (1-плюс,-1-минус),количество разрядов в целой части,количество разрядов в дробной части элемента в первой строчке и первом столбике матрицы исследования,
- в следующей строчке через пробел идут разряды целой части элемента в первой строчке и первом столбце матрицы исследования,
- в следующей строчке через пробел идут разряды дробной части элемента в первой строчке и первом столбце матрицы исследования,

- в следующей строчке идут через пробел знак (1-плюс,-1-минус), количество разрядов в целой части, количество разрядов в дробной части элемента в первой строчке и втором столбике матрицы исследования,
- в следующей строчке через пробел идут разряды целой части элемента в первой строчке и втором столбце матрицы исследования,
- в следующей строчке через пробел идут разряды дробной части элемента в первой строчке и втором столбце матрицы исследования,
- далее в построчной структуре идут в аналогичном виде все элементы матрицы исследования(приложение-2).

Результат работы программы `dlmnfraz` будет в файле `dlmnfraz` и в файле `rowfraz`(проверка решения систем нормальных уравнений).

Результат работы программы `dlmncraz` будет в файле `dlmncraz` и в файле `rowcraz`(проверка решения систем нормальных уравнений)(приложение-2).

Все выходные файлы смотрим любым текстовым редактором.

### **1.10 Поставка программного комплекса**

В поставку программ по реализации метода наименьших квадратов в длинной арифметике входят:

1. `dlmnkfor.arj`-архив по реализации в Фортране MS/DOS
2. `dlmnkcpp.arj`-архив по реализации в BORLAND C++ в WINDOWS.
3. `dmnkopis.arj`-архив по описанию реализации метода наименьших квадратов в длинной арифметике.

## Глава -2 Алгоритмическое описание программного комплекса (язык Фортран)

### 2.1 Подпрограмма по реализации сложения без учета знаков слагаемых

Вызов подпрограммы(приложение-1):

**call sl (n, nd, a, ad, m, md, b, bd, kcel, kdro, rezc, rezdr, krazr),**

n – количество разрядов в целой части первого слагаемого,

nd – количество разрядов в дробной части первого слагаемого,

a – массив разрядов целой части первого слагаемого,

ad - массив разрядов дробной части первого слагаемого,

m - количество разрядов в целой части второго слагаемого,

md - количество разрядов в дробной части второго слагаемого,

b – массив разрядов целой части второго слагаемого,

bd – массив разрядов дробной части второго слагаемого,

kcel – количество разрядов в целой части суммы,

kdro - количество разрядов в дробной части суммы,

rezc – массив разрядов целой части суммы,

rezdr - массив разрядов дробной части суммы,

krazr - количество разрядов в дробной части длинной арифметики.

Здесь и далее все массивы-целые двухбайтовые, скалярные переменные-целые четырехбайтовые.

В цикле do 66875 происходит обнуление массивов целой и дробной части суммы. В операторе if(n.ge.m) goto 897 идет сравнение длин целых частей слагаемых.

Если длина целой части первого слагаемого меньше длины целой части второго слагаемого, то начиная с оператора do 678 и по метку 897 идет пересылка целых и дробных частей слагаемых.

После метки 897 идет нахождение максимальной длины дробной части слагаемых, расширение в цикле do 79 дробной части второго слагаемого нулями до максимальной длины дробной части, расширение в цикле do 82 дробной части первого слагаемого нулями до максимальной длины дробной части.

После метки 81 в цикле do 100 идет формирование единого массива разрядов  $a(i)$  для первого слагаемого (разряды целой части + разряды дробной части).

В цикле do 101 идет формирование единого массива разрядов  $b(i)$  для второго слагаемого (разряды целой части + разряды дробной части).

Далее идет непосредственное поразрядное сложение.

При сложении:

$i$  – счетчик разрядов в сумме,

$r$  – значение переноса из младшего разряда в старший,

$nt$  – текущий номер разряда в первом слагаемом, начинается с младшего разряда,

$mt$  – текущий номер разряда во втором слагаемом, начинается с младшего разряда.

После метки 7 идет поразрядное суммирование элементов массивов в двух слагаемых.

В переменной  $i1$  будет значение переноса из младшего разряда в следующий старший.

В переменной OS - результат суммирования разрядов.

Этот результат суммирования помещается в элемент  $s(i)$  массива результата суммирования.

После уменьшения номеров разрядов идет анализ текущих номеров разрядов слагаемых и значения переноса  $P$ .

После метки 67:

ik – номер младшего разряда в целой части суммы,

kcel – количество разрядов в целой части суммы,

kdro – количество разрядов в дробной части суммы.

В цикле do 1121 идет пересылка в массив rezc целой части суммы (последний из массива «с» будет первым в массиве rezc).

В цикле do 1122 идет пересылка в массив rezdr дробной части суммы (последний из дробной части будет первым в rezdr).

Далее идет анализ длин целой и дробной части результата суммирования.

## **2.2 Подпрограмма по реализации сложения с учетом знаков слагаемых**

Вызов подпрограммы(приложение-1):

**call\_slzn(za, zb, zc, n, nd, a, ad, m, md, b, bd, kcel, kdro, rezc, rezdr, krazr),**

za – знак первого слагаемого,

zb – знак второго слагаемого,

zc – знак суммы,

остальные параметры аналогичны параметрам при вызове подпрограммы Sl.

В подпрограмме slzn идет сначала запоминание исходных значений слагаемых.

В переменной ip – произведение знаков слагаемых.

Далее идет анализ знаков слагаемых, длин целых частей слагаемых.

По результатам анализа идет выбор операций.

В цикле do 786 идет анализ целых частей слагаемых при равенстве длин целых частей.

Далее идет нахождение минимальной длины дробной части двух слагаемых.

В цикле do 791 идет анализ дробных частей слагаемых при равенстве целых частей слагаемых.

В операторе if(nd.le.md) goto 795 идет анализ длин дробных частей слагаемых.

Если первое слагаемое положительное, то после метки 782 идут пересылки:

$a \leftrightarrow b$ ,

$ad \leftrightarrow bd$ ,

$n \leftrightarrow m$ ,

$nd \leftrightarrow md$ ,

$za \leftrightarrow zb$

После метки 785 идет восстановление исходных значений слагаемых.

### **2.3 Подпрограмма по реализации вычитания без учета знаков уменьшаемого и вычитаемого**

Вызов подпрограммы(приложение-1):

**call wich (n,nd,a,ad,m,md,b,bd,kcra,kdra,cra,dra,krazr),**

n- количество разрядов в целой части уменьшаемого,

nd- количество разрядов в дробной части уменьшаемого,

a - массив разрядов целой части уменьшаемого,

ad - массив разрядов дробной части уменьшаемого,

m - количество разрядов в целой части вычитаемого,

md – количество разрядов в дробной части вычитаемого,

b – массив разрядов целой части вычитаемого,

bd – массив разрядов дробной части вычитаемого,

kcra - количество разрядов в целой части разности,

kdra – количество разрядов в дробной части разности,

cra – массив разрядов целой части разности,

dra – массив разрядов дробной части разности,

krazr - количество разрядов в дробной части длинной арифметики.

В цикле do 56881 идет обнуление массивов целой и дробной частей разности, результата вычитания.

В операторе if(n.ge.m) goto 897 идет сравнение длин целых частей уменьшаемого и вычитаемого.

Если количество разрядов в целой части уменьшаемого меньше чем количество разрядов в целой части вычитаемого, то осуществляем пересылки целых и дробных частей уменьшаемого и вычитаемого:

$a \leftrightarrow b$ ,

$ad \leftrightarrow bd$ ,

$n \leftrightarrow m$ ,

$nd \leftrightarrow md$

Реализация пересылок с оператора do 678 по метку 897.

После метки 897 находим максимальное количество разрядов для дробной части результата вычитания, разности.

В цикле do 79 идет расширение дробной части вычитаемого.

В цикле do 82 идет расширение дробной части уменьшаемого.

В цикле do 100 идет формирование единого массива уменьшаемого (целая + дробная часть).

В цикле do 101 идет формирование единого массива вычитаемого (целая + дробная часть).

Далее в подпрограмме:

i – номер разряда результата вычитания, разности,

$nt$  – номер разряда в уменьшаемом,

$mt$  – номер разряда в вычитаемом.

В операторе `if(a(nt).ge.b(mt)) goto 1` идет сравнение разрядов в уменьшаемом и вычитаемом.

С метки 3 по оператор `goto 3` идет поиск ненулевого разряда, который старше разряда с номером  $nt$  в массиве «а» (это когда  $a(nt) < b(mt)$ ).

После метки 2 в операторе

$a(np) = a(np) - 1$

идет уменьшение ненулевого разряда в массиве «а» на 1 (занимание единицы при вычитании).

С метки 7 по оператор `goto 7` идет заполнение нулевых разрядов в массиве «а» от  $np+1$  разряда до  $nt-1$  разряда девятками.

После метки 30 идет прибавление 10 к разряду  $a(nt)$  в уменьшаемом.

После метки 1 идет увеличение счетчика разрядов в результате вычитания, разности, и присваивание

$c(i) = a(nt) - b(mt)$

Далее уменьшение на 1 номеров разрядов  $nt$ ,  $mt$ , анализ значений  $mt, n, m$ , заполнение массива результата вычитания разрядами массива «а», когда их больше по количеству чем в массиве «b» (количество разрядов в целых частях).

После метки 999:

$kсга$  – количество разрядов в целой части разности,

$kdra$  – количество разрядов в дробной части разности,

$im$  – количество ведущих нулей в разности (0001).

С оператора `do 2001` по метку 973 идет формирование массива  $сга$  целой части результата вычитания.

После метки 973

$kсга = kсга - im$  - количество разрядов в целой части разности после исключения нулей в старших разрядах.

В цикле do 2002 идет формирование массива разрядов дробной части разности dra.

Далее идет уточнение количества разрядов в целой и дробной частях результата вычитания.

#### **2.4 Подпрограмма по реализации вычитания с учетом знаков уменьшаемого и вычитаемого**

Вызов подпрограммы(приложение-1):

**call wiczn(za, zb, zc, n, nd, a, ad, m, md, b, bd, kcel, kdra, rezc, rezdr, krazr),**

za – знак уменьшаемого,

zb – знак вычитаемого,

zc – знак разности,

остальные параметры аналогичны параметрам подпрограммы wicb при вызове.

Сначала в подпрограмме wiczn происходит запоминание исходных значений уменьшаемого и вычитаемого.

Потом в операторе

$zb=zb \times (-1)$

происходит смена знака вычитаемого.

После этого далее по алгоритму подпрограмма wiczn полностью аналогична подпрограмме slzn.

## 2.5 Подпрограмма по реализации умножения без учета знаков

Вызов подпрограммы(приложение-1):

**call ymn (n, nd, a, ad, m, md, b, bd, kcel, kdro, ctot, cdtot, krazr),**

n – количество разрядов в целой части первого сомножителя,  
 nd - количество разрядов в дробной части первого сомножителя,  
 a – массив разрядов целой части первого сомножителя,  
 ad – массив разрядов дробной части первого сомножителя,  
 m – количество разрядов в целой части второго сомножителя,  
 md – количество разрядов в дробной части второго сомножителя,  
 b – массив разрядов целой части второго сомножителя,  
 bd – массив разрядов дробной части второго сомножителя,  
 kcel – количество разрядов в целой части произведения,  
 kdro – количество разрядов в дробной части произведения,  
 ctot – массив разрядов целой части произведения,  
 cdtot – массив разрядов дробной части произведения,  
 krazr – количество разрядов в дробной части длинной арифметики.

В цикле do 66987 идет обнуление целой и дробной частей сумматора.

В цикле do 100 идет формирование массива «a» для первого сомножителя (целая + дробная часть).

В цикле do 101 идет формирование массива “b” для второго сомножителя (целая + дробная часть).

Далее:

ntot – количество разрядов в целой части сумматора,

kdtot - количество разрядов в дробной части сумматора.

В цикле do 651 идет обнуление целой части сумматора.

В цикле `do 652` идет обнуление дробной части сумматора.

Далее:

`i` – счетчик разрядов в произведении,

`im` – счетчик количества разрядов для определения конца дробной части произведения и начала целой части произведения,

`r` – значение переноса из младшего разряда в старший разряд,

`nt` – номер разряда в массиве «a» первого сомножителя,

`mt` – номер разряда в массиве «b» второго сомножителя.

В цикле `do 941` идет обнуление результата произведения всех разрядов первого сомножителя на один разряд второго сомножителя.

С метки 701 по метку 1701 идет поразрядное умножение первого сомножителя «a» (все разряды) на один разряд `b(mt)` второго сомножителя.

Результат произведения в массиве `c(i)`.

После метки 1701 и до метки 8722 идет формирование целой (массив `cs1`) и дробной (массив `csld`) частей результата умножения первого множителя (всех разрядов) на разряд второго множителя.

После метки 8722 подпрограмма `sl` осуществляет суммирование результатов умножения (массивы `cs1`, `csld`) первого множителя (всех разрядов) на разряд второго множителя.

В качестве второго слагаемого в `sl` идет сумматор (массивы `ctot`, `cdtot`).

Результат суммирования в массивах `rezc`, `rezdr`.

После суммирования идет переопределение сумматора в циклах `do 704` и `do 705`.

Далее изменяются параметры сумматора `ntot`, `kdtot`.

После этого идет уменьшение номера разряда во втором сомножителе и так в цикле для всех разрядов второго сомножителя осуществляется их умножение на первый сомножитель (все разряды) и суммация этих произведений.

Результат произведения сомножителей будет в массивах `ctot`, `cdtot` с параметрами `kcel` (длина целой части произведения), `kdro` (длина дробной части произведения).

Далее в циклах `do 7766` и `do 7768` идет формирование целой части результата умножения.

Убираются ведущие нули 0001.

Далее идет уточнение количества разрядов в целой и дробной частях произведения.

## 2.6 Подпрограмма умножения с учетом знаков сомножителей

Вызов подпрограммы(приложение-1):

**call ymnzn (za, zb, zc, n, nd, a, ad, m, md, b, bd, kcel, kdro, ctot, cdtot, krazr),**

`za` – знак первого сомножителя,

`zb` – знак второго сомножителя,

`zc` – знак произведения.

Остальные параметры полностью аналогичны параметрам при вызове подпрограммы `umn`.

Сначала в подпрограмме идет запоминание исходных значений сомножителей.

Потом определение знака произведения:

`zc=za×zb`

Далее идет вызов подпрограммы `umn` по реализации длинного умножения.

Потом идет восстановление исходных значений сомножителей.

## 2.7 Подпрограмма по реализации деления без учета знаков

Вызов подпрограммы(приложение-1):

**call deld (n, nd, a, ad, m, md, b, bd, kcdel, kddel, cdel, ddel, krazr),**

n – количество разрядов в целой части делимого,

nd – количество разрядов в дробной части делимого,

a – массив разрядов целой части делимого,

ad – массив разрядов дробной части делимого,

m – количество разрядов в целой части делителя,

md – количество разрядов в дробной части делителя,

b – массив разрядов целой части делителя,

bd – массив разрядов дробной части делителя,

kcdel – количество разрядов в целой части частного,

kddel – количество разрядов в дробной части частного,

cdel – массив разрядов целой части частного,

ddel – массив разрядов дробной части частного,

krazr – количество разрядов в дробной части длинной арифметики.

В цикле `do 66789` идет обнуление результатов деления, массивов `cdel`, `ddel`.

Далее находится максимальная длина для дробной части.

В цикле `do 79` идет добавление нулей в дробной части делителя.

В цикле `do 82` идет добавление нулей в дробной части делимого.

В цикле `do 100` формируем массив “a” делимого (целая + дробная часть).

В циклах `do 5671` и `do 5673` убираем ведущие нули в массиве “a” (~~000~~1).

В циклах `do 101`, `do 15671`, `do 15673` аналогично идет формирование массива “b”, делителя.

В цикле do 2677 идет обнуление сумматора, массивы ctot, cdtot.

Далее:

kctot – количество разрядов в целой части сумматора,

kdto – количество разрядов в дробной части сумматора,

се – целая часть единичного значения,

kedі - количество разрядов в целой части единичного значения,

сед – дробная часть единичного значения,

kedid – количество разрядов в дробной части единичного значения,

kad – количество разрядов в дробной части делимого после преобразования его в один массив “а” (целая + дробная часть исходного значения),

kbd – количество разрядов в дробной части делителя после преобразования его в один массив “b” (целая + дробная часть исходного значения),

ad – дробная часть делимого после его преобразования в один массив “а” (целая + дробная часть исходного значения),

bd – дробная часть делителя после его преобразования в один массив “b” (целая + дробная часть исходного значения),

nat – количество разрядов в делимом,

mat – количество разрядов в делителе,

isd – счетчик разрядов в дробной части частного,

kcel – количество разрядов в целой части результата суммирования (подпрограмма sl),

kdro – количество разрядов в дробной части результата суммирования (подпрограмма sl),

rezc – массив разрядов целой части результата суммирования (подпрограмма sl),

rezdr – массив разрядов дробной части результата суммирования (подпрограмма sl).

С метки 2010 по метку 1231 реализуется алгоритм вычитания из делимого делителя и подсчета количества этих вычитаний, то есть расчет значения частного.

После метки 1231 идет корректировка количества разрядов в целой и дробной частях частного.

## 2.8 Подпрограмма по реализации деления с учетом знаков

Вызов подпрограммы(приложение-1):

**call delzn (za, zb, zc, n, nd, a, ad, m, md, b, bd, kcdel, kddel, cdel, ddel, krazr),**

za – знак делимого,

zb – знак делителя.

zc – знак частного,

остальные параметры аналогичны параметрам подпрограммы deld при вызове.

Вначале подпрограммы идет запоминание исходных данных, значений делимого и делителя.

Потом идет вычисление знака частного:

$$zc=za \times zb$$

Потом идет вызов подпрограммы deld для реализации длинного деления.

Потом идет восстановление исходных данных, делимого и делителя.

## 2.9 Подпрограмма по реализации прямого хода в методе Гаусса

Вызов подпрограммы(приложение-1):

**call gaydl (n, znak, nc, nd, ac, ad, znp, nbc, nbd, bc, bd, znx, nxc, nxd, xc,  
xd, igau, krazr),**

n – порядок системы,

znak – массив знаков элементов матрицы системы,

nc– массив количества разрядов целых частей элементов матрицы системы,

nd – массив количества разрядов дробных частей элементов матрицы системы,

ac – массив разрядов целых частей элементов матрицы системы,

ad – массив разрядов дробных частей элементов матрицы системы,

znp – массив знаков свободных членов ,

nbc –массив длин целых частей свободных членов,

nbd –массив длин дробных частей свободных членов,

bc – массив разрядов целых частей свободных членов ,

bd – массив разрядов дробных частей свободных членов,

znx – массив знаков неизвестных,

nxc –массив длин целых частей неизвестных,

nxd –массив длин дробных частей неизвестных,

xc –массив разрядов целых частей неизвестных,

xd(i,l) – массив разрядов дробных частей неизвестных,

igau – индикатор ошибки,

krazr – количество разрядов в дробной части длинной арифметики.

В цикле do 38247 идет заполнение массива mpreг номерами неизвестных.

Массив `mpreg` необходим для сохранения исходных номеров неизвестных при выборе главного элемента в методе Гаусса.

Далее идет анализ количества разрядов в дробной части длинной арифметики.

В переменной `pis` запоминаем исходное значение порядка системы.

$i$  – номер уравнения, которое вычитается в прямом ходе метода Гаусса,

$j$  – номер неизвестной, номер столбца в матрице системы.

Подпрограмма `pitax` находит главный элемент и помещает его в  $i$ -ую строчку и  $j$ -ый столбец, осуществляя при этом перестановку строк, уравнений, и столбцов, неизвестных.

Далее с `nct=nc(i,i)` до оператора `do 41028` идет определение  $t=a(i,i)$  в прямом ходе Гаусса.

С оператора `do 41028` до оператора `do 41029` идет определение нулевого значения для длинной арифметики.

С оператора `do 41029` до оператора  $l=i+1$  идет сравнение  $/t/$  (модуль) с нулем.

В случае нулевого значения  $t$  далее с оператора  $l=1+i$  по оператор `if(1.le.n) goto 4` идет отыскание ненулевого элемента в  $i$ -ом столбце.

В случае не нахождения ненулевого элемента в  $i$ -ом столбце индикатор ошибки `igau=0` и решение завершается.

Если ненулевой элемент найден в столбце  $i$ , то с метки 3 по метку 5 осуществляется перестановка элементов  $i$ -ой и  $l$ -ой строк в матрице системы.

С метки 5 по метку 41045 включительно осуществляется перестановка правых частей  $i$ -ого и  $l$ -ого уравнений.

Далее осуществляется определение ненулевого  $t=a(i,i)$ .

С метки 2222 по оператор `if(j.le.n) goto 2222` идет деление левой части  $i$ -ого уравнения на диагональный элемент этого уравнения.

Далее до метки 41055 включительно происходит деление правой части  $i$ -ого уравнения  $b(i)=b(i)/t$ .

Переменная  $k$  – это номер уравнения, из которого идет вычитание  $i$ -ого уравнения в прямом ходе метода Гаусса.

После метки 7111 по метку 41057 включительно определяется  $p=a(k,i)$ .

С метки 7 по метку 41061 включительно определяется  $p \times a(i,j)$

Далее до метки 41065 идет вычисление  $a(k,j)=a(k,j)- p \times a(i,j)$ .

Далее  $j$  увеличивается на единицу и процедура вычитания идет для следующего элемента в  $k$ -ой строке.

После вычитания левых частей  $k$ -ого и  $i$ -ого уравнений далее идет вычитание правых частей  $k$ -ого и  $i$ -ого уравнений.

Далее  $k$  увеличивается на единицу и процедура вычитания  $i$ -ого уравнения идет для следующего уравнения системы.

После завершения процедуры вычитания для  $i$ -ого уравнения значение  $i$  увеличивается на единицу и процедура вычитания идет для следующего уравнения системы.

После завершения прямого хода идет вызов подпрограммы `obrdl` по реализации обратного хода.

После обратного хода  $n=nis$ - восстановили исходное значение порядка системы.

## 2.10 Подпрограмма по нахождению главного элемента

Вызов подпрограммы(приложение-1):

**call nimax (i, j, n, nc, nd, znak, ac, ad, nbc, nbd, znp, bc, bd, mnper, krazr),**

$i, j$  – номер строки и номер столбца элемента в матрице системы, с которого начинается поиск главного элемента,

$n$  – порядок системы,

$nc$  – массив длин целых частей элементов матрицы системы,  
 $nd$  – массив длин дробных частей элементов матрицы системы,  
 $znak$  – массив знаков элементов матрицы системы,  
 $ac$  – массив разрядов целых частей элементов матрицы системы,  
 $ad$  – массив разрядов дробных частей элементов матрицы системы,  
 $nbc$  – массив длин целых частей свободных членов системы,  
 $nbd$  – массив длин дробных частей свободных членов системы,  
 $znp$  – массив знаков свободных членов системы,  
 $bc$  – массив разрядов целых частей свободных членов системы,  
 $bd$  – массив разрядов дробных частей свободных членов системы,  
 $mpreg$  – массив номеров неизвестных системы.

Далее, начиная с элемента  $c_{ij}$ , находится максимальный по модулю элемент матрицы системы. Циклы `do 38250, do 38250`.

Координаты этого элемента:

$imax$  – номер строки максимального элемента,

$jmax$  – номер столбца максимального элемента.

Переменные:

$kmax$  – количество разрядов в целой части максимального элемента,

$kdmax$  – количество разрядов в дробной части максимального элемента.

Массивы:

$xmaxc$  – массив разрядов целой части максимального элемента,

$xmaxd$  – массив разрядов дробной части максимального элемента.

С цикла `do 38251` до цикла `do 38252` идет пересылка строки  $i$  со строкой  $imax$ , пересылка свободного члена с номером  $i$  со свободным членом под номером  $imax$ .

В цикле `do 38252` идет пересылка столбца  $j$  со столбцом  $jmax$ .

Далее идет пересылка в массиве номеров неизвестных  $mpreg$ .

## 2.11 Подпрограмма по реализации обратного хода в методе Гаусса

Вызов подпрограммы(приложение-1):

**call obrdl (n, znak, nc, nd, ac, ad, znp, nbc, nbd, bc, bd, znx, nxc, pxd, xc,  
xd, igau, mnper, anc, andr, krazr),**

n – порядок системы,

znak – массив знаков элементов матрицы системы,

nc – массив длин целых частей элементов матрицы системы,

nd - массив длин дробных частей элементов матрицы системы,

ac - массив разрядов целых частей элементов матрицы системы,

ad - массив разрядов дробных частей элементов матрицы системы,

znp – массив знаков правых частей системы,

nbc – массив длин целых частей свободных членов системы,

nbd - массив длин дробных частей свободных членов системы,

bc - массив разрядов целых частей свободных членов системы,

bd - массив разрядов дробных частей свободных членов системы,

znx – массив знаков неизвестных системы,

nxc - массив длин целых частей неизвестных системы,

pxd - массив длин дробных частей неизвестных системы,

xc - массив разрядов целых частей неизвестных системы,

xd - массив разрядов дробных частей неизвестных системы,

igau – код ошибки,

mnper - массив номеров неизвестных системы,

anc - массив разрядов целой части нулевого значения,

andr - массив разрядов дробной части нулевого значения,

krazr – количество разрядов в дробной части длинной арифметики.

В переменной  $nis$  запоминаем исходное значение порядка матрицы системы.

С оператора  $nct=nc(n,n)$  по оператор  $do$  41077 идет сравнение элемента  $a(n,n)$  с нулем.

Если элемент  $a(n,n)$  нулевой, то код ошибки  $igau=0$  и процесс вычислений завершаем.

Если элемент  $a(n,n)$  имеет ненулевое значение, то переходим к метке 82.

В переменной  $nt$  запоминаем исходное значение порядка системы  $n$  (далее  $n$  будет изменяться).

С оператора  $ncr=nbc(n)$  по оператор  $do$  41091 включительно идет вычисление  $x(n)=b(n)/a(n,n)$ .

С 85 метки идет цикл по нахождению неизвестных в обратном ходе метода Гаусса.

Переменные:

$ncsum$  – количество разрядов в целой части сумматора,

$ndsum$  – количество разрядов в дробной части сумматора,

$izsum$  – знак сумматора,

$icsum$  – массив разрядов в целой части сумматора,

$idsun$  – массив разрядов в дробной части сумматора,

$j$  – номер уже найденной неизвестной,

С метки 84 по оператор  $do$  41098 включительно идет вычисление  $a(n,j) \times x(j)$ .

Далее по оператор  $do$  41100 включительно идет вычисление

$sum=sum+a(n,j)*x(j)$ .

Далее  $j$  увеличивается на единицу и в цикле (с метки 84) идет опять процесс суммирования.

После вычисления суммы парных произведений на уже найденных неизвестных с оператора  $ncr=nbc(n)$  по оператор  $do$  41104 включительно идет вычисление неизвестной  $x(n)=b(n)-sum$ .

После этого переходим на метку 85 и уменьшаем в цикле номер уравнения, номер неизвестной  $n=n-1$ .

После прохода всех уравнений (с последнего до первого) уходим на метку 83,  $igau = 1$ .

В цикле do 38254 определяем вектор  $b(j)=x(i)$ .

В цикле do 38255 определяем неизвестные под исходными номерами  $x(i)=b(i)$ .

В операторе  $n=nis$  определяем исходное значение порядка системы.

## 2.12 Подпрограмма по проверке решения системы методом Гаусса

**call prowg (n, znak, nc, nd, ac, ad, znp, nbc, nbd, bc, bd, znx, pxc, pxd, xc, xd, igau, krazr),**

$n$  – порядок системы,

$znak$  – массив знаков элементов матрицы системы,

$nc$  – массив длин целых частей элементов матрицы системы,

$nd$  – массив длин дробных частей элементов матрицы системы,

$ac$  – массив разрядов целых частей элементов матрицы системы,

$ad$  – массив разрядов дробных частей элементов матрицы системы,

$znp$  – массив знаков правых частей системы,

$nbc$  – массив длин целых частей свободных членов системы,

$nbd$  – массив длин дробных частей свободных членов системы,

$bc$  – массив разрядов целых частей свободных членов системы,

$bd$  – массив разрядов дробных частей свободных членов системы,

$znx$  – массив знаков неизвестных системы,

$pxc$  – массив длин целых частей неизвестных системы,

$pxd$  – массив длин дробных частей неизвестных системы,

xc – массив разрядов целых частей неизвестных системы,

xd – массив разрядов дробных частей неизвестных системы,

igau – код ошибки,

krazr – количество разрядов в дробной части в длинной арифметике.

Осуществляется ввод порядка системы, количества разрядов в дробной части в длинной арифметике.

Далее вводятся и записываются элементы матрицы системы и свободные члены системы.

Далее записываются неизвестные системы.

В цикле до 231 осуществляется вычисление правых частей системы и запись исходных значений правых частей, посчитанных правых частей и разницы между ними.

### **2.13 Подпрограмма mnk**

В подпрограмме(приложение-1) осуществляется построение системы нормальных уравнений в методе наименьших квадратов, вызов подпрограмм gaydl (прямой ход метода Гаусса), growg (проверка решения).

### **2.14 Главная программа dlmnk2**

Осуществляется(приложение-1) ввод матрицы исследования, названий параметров-столбиков, номеров параметров-аргументов, начальный и конечный номер зависимого аргумента, порядка аппроксимирующего полинома, количества разрядов в дробной части в длинной арифметике, количества разрядов в целой и дробной частях при выделении памяти, вызов подпрограммы mnk.

**СПИСОК ЛИТЕРАТУРЫ**

1. Й.Ленгсам, М.Огенстайн, А.Тененбаум  
Структуры данных для персональных ЭВМ//Издательство  
"Мир", Москва,1989.
2. А.Aho, J.Норcroft, J.Ullman  
Data Structures and Algorithms//Addison-Wesley, Reading, Mass, 1982.
3. Ван Тассел Д.  
Стиль, разработка, эффективность, отладка и испытание  
программ // Издательство "Мир", Москва, 1981.
4. Дал У., Дейкстра Э., Хоор К.  
Структурное программирование// М.: Мир, 1975.
5. Ахо А., Хопкрофт Дж., Ульман Дж.  
Построение и анализ вычислительных алгоритмов//М.:Мир, 1979.
6. Р.Джордейн  
Справочник программиста персональных компьютеров типа  
IBM PC, XT и AT//Москва,"Финансы и статистика", 1992.
7. Фирма "Диалектика" TURBO VISION для C++ // Киев, 1992.
8. Мак-Кракен, Дорн  
Численные методы и программирование на Фортране //  
Издательство "Мир", Москва, 1982.
9. Д.Стэбли  
Логическое программирование в системе IBM/360. //  
Издательство "Мир", Москва, 1979.
10. У.Радд  
Программирование на языке ассемблера и вычислительные  
системы IBM 360 и 370 // Издательство "Мир", Москва,1979.
11. Дж.Донован  
Системное программирование // Москва,"Мир",1976.

## ПРИЛОЖЕНИЕ-1(наборы с исходными текстами программ на Фортране,исходные и выходные данные для программы на Фортране)

### Набор dlmnk2.for

\$large

с-метод наименьших квадратов в длинной арифметике

```

с      dimension x(100,20),y(100),akm(10,100),mnarg(20),xr(100)
с      dimension ym(100),ak(100),delm(10),zgl(80),rab(100,150)
с      dimension anaz(150,60),wkl(20)
с      double precision ym,dx,x,y,ak,akm,delm,delt,rab,xr,wkl,wkltot,sy
с      double precision xmin
      dimension anaz(20,60),mnarg(20),zgl(80)
      integer*2 izrab(50,20),kcrab(50,20),kdrab(50,20),
*icrab(50,20,20),idrab(50,20,20),izy(50),kcy(50),kdy(50),
*icy(50,20),idy(50,20),izx(50,10),kcx(50,10),kdx(50,10),
*icx(50,10,20),idx(50,10,20),izakm(3,31),izak(31),kcakm(3,31),
*kcak(31),kdakm(3,31),kdak(31),icakm(3,31,20),icak(31,20),
*idakm(3,31,20),idak(31,20)
      integer*2 izdelm(3),kcdelm(3),kddelm(3),icdelm(3,20),
*iddelm(3,20),izym(50),kcy(50),kdym(50),icy(50,20),idym(50,20),

*icw(400),idw(400),icr(400),idr(400),icrez(400),idrez(400),
*icl(400),
*idl(400),icn(400),idn(400),icxmi(400),idxmi(400),icdelt(400),
*iddelt(400),izwkl(10),kcwkl(10),kdwkl(10),icwkl(10,20),
*idwkl(10,20)
      integer*2 icsy(400),idsy(400),icwtot(400),idwtot(400)
      double precision rwwod(20)
      open(4,file='sled',status='old')
      open(6,file='dlmnk2',status='Unknown')
      open(5,file='prowfdl2',status='Unknown')
      open(7,file='name',status='old')
с      open(9,file='prompr2',status='Unknown')
      read(7,*) knaz
      if(knaz.gt.20) goto 1762
      do 1441 i=1,knaz
      do 83 j=1,60
      anaz(i,j)= ' '
83      continue
      read(7,1442) (anaz(i,j),j=1,60)
1442 format(60a1)
1441 continue

      read(4,1001) zgl
1001 format(80a1)
      write(6,1001) zgl
      read(4,*) n,m
      if(n.le.50) goto 1443
      write(6,*) 'число строчек-наблюдений должно быть<=50'
      stop
1443 continue
      if(m.le.20) goto 1444
1762 continue
      write(6,*) 'число параметров-столбиков должно быть<=20'
      stop
1444 continue
      nt=n
      l=0
11601 continue
      ios=nt-nt/10*10

      l=l+1

```

```

    id1(1)=ios
    nt=nt/10
    if(nt.gt.0) goto 11601
    izn=1
    kcn=1
    kdn=1
    lt=1
    do 11602 l=1,kcn
    icn(1)=id1(lt)
    lt=lt-1
11602 continue
    idn(1)=0

с-ввод исходной матрицы в длинном формате
    do 1002 i=1,n
с    do 11001 j=1,m
    read(4,*) (rwwod(j),j=1,m)
    do 11001 j=1,m
        icrww=dabs(rwwod(j))
    idrww=dabs((dabs(rwwod(j))-icrww)*(1.d+1)**9)
    izr=1
    if(rwwod(j).ge.0) goto 14081
    izr=-1
14081 continue
    nt=icrww
    l=0
14082 continue
    ios=nt-nt/10*10
    l=l+1
    id1(1)=ios
    nt=nt/10
    if(nt.gt.0) goto 14082
    izrab(i,j)=izr
    kcrab(i,j)=1
    kcel=1
    lt=1
    iz1=1
    do 14083 l=1,iz1
    icrab(i,j,l)=id1(lt)
    lt=lt-1
14083 continue

    nt=idrww
    l=0
14084 continue
    ios=nt-nt/10*10
    l=l+1
    id1(1)=ios
    nt=nt/10
    if(nt.gt.0) goto 14084
    if(l.eq.9) goto 37601
    k1=9-l
    do 37602 j1=1,k1
    l=l+1
    id1(1)=0
37602 continue
37601 continue
    kdrab(i,j)=1
    lt=1
    iz1=1
    kdel=1
    do 14085 l=1,iz1
    idrab(i,j,l)=id1(lt)

```

```

        lt=lt-1
14085 continue
        write(6,*) 'целая часть'
        write(6,19781) (icrab(i,j,l),l=1,kcel)
19781 format(1x,50i1)
        write(6,*) 'дробная часть'
        write(6,19781) (idrab(i,j,l),l=1,kdel)
11002 format(70i1)
11001 continue
c      read(4,*) (rab(i,j),j=1,m)
c      write(6,*) (rab(i,j),j=1,m)
1002  continue
7775  continue
c      stop
c      write(*,*) 'введите максимальную степень<=10'
c      read(*,*) kmax
c      write(*,*) 'введите начальную степень'
c      read(*,*) kna
        kna=1
        k=kna
        write(*,*) 'Введите количество параметров-аргументов<=10'
        read(*,*) karg
c      kmax=100/karg-1
c      if(kmax.gt.10) kmax=6
c      kmax=4
        write(*,*) 'введите через пробел номера аргументов'
        read(*,*) (mnarg(i),i=1,karg)
        write(*,1003)
1003  format(1x,'Введите через пробел начальный и конечный номер'/1x,
* 'зависимой переменной')
        read(*,*) inay,ikoу
        write(*,*) 'введите степень полинома<=3'
        read(*,*) kmax
        write(*,*) 'введите количество разрядов для арифметики<=20'
        read(*,*) krazr
        write(6,13667) kmax
13667 format(//1x,'степень полинома=',i2//)
        m=karg
        do 2001 ipу=inay,ikoу
            k=kna
            write(6,2003) ipу,(anaz(ipу,jl),jl=1,60)
            write(6,*) 'номера аргументов'
            write(6,2004) (mnarg(i),i=1,karg)
403  continue
c-формирование вектора у для длинной арифметики
        do 2002 i=1,n
            izy(i)=izrab(i,ipу)
            kcy(i)=kcrab(i,ipу)
            kdy(i)=kdrab(i,ipу)
            kcel=kcy(i)
            kdel=kdy(i)
            do 11021 l=1,kcel
                icy(i,l)=icrab(i,ipу,l)
11021 continue
            do 11022 l=1,kdel
                idy(i,l)=idrab(i,ipу,l)
11022 continue
c      y(i)=rab(i,ipу)
2002  continue
2003  format(1x,'зависимый параметр-',i3/1x,60a1/)
2004  format(1x,15i5)
c      write(6,*) 'исходный зависимый вектор'
c      write(6,*) (y(i),i=1,n)

```

с-формирование матрицы x аргументов для длинной арифметики

```

с      pause 'x'
      ila=0
      do 9001 iarg=1,karg
        narg=mnarg(iarg)
с      write(6,1231) narg
1231   format(1x,'arg=',i3)
      ila=ila+1
      do 9002 i=1,n
        izx(i,ila)=izrab(i,narg)
        kcx(i,ila)=kcrab(i,narg)
        kdx(i,ila)=kdrab(i,narg)
        kcel=kcx(i,ila)
        kdel=kdx(i,ila)
        do 11031 l=1,kcel
          icx(i,ila,l)=icrab(i,narg,l)
11031  continue
        do 11032 l=1,kdel
          idx(i,ila,l)=idrab(i,narg,l)
11032  continue
с      x(i,ila)=rab(i,narg)
9002   continue
9001   continue
с      write(6,*) 'matrix x'
      do 9004 i=1,n
с      write(6,8771) (x(i,j),j=1,karg)
8771   format(1x,5f13.4)
9004   continue
с      write(6,*) 'зависимый вектор'
с      write(6,8771) (y(i),i=1,n)
с      pause 'pered mnk'
      call mnk(n,karg,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
*izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,igau,krazr)
с      pause 'posle mnk'
с      write(*,*) igau
      if(igau.eq.0) goto 401

      km1=k*m+1
с      write(6,123) k
с      write(6,124) (ak(ip),ip=1,km1)
124    format(1x,6f13.5)
123    format(1x,'k=',i3/1x,6f12.5)
      do 8 i=1,km1
        izakm(k,i)=izak(i)
        kcakm(k,i)=kcak(i)
        kdakm(k,i)=kdak(i)
        kcel=kcak(i)
        kdel=kdak(i)
        do 11041 l=1,kcel
          icakm(k,i,l)=icak(i,l)
11041  continue
        do 11042 l=1,kdel
          idakm(k,i,l)=idak(i,l)
11042  continue
с      akm(k,i)=ak(i)
8      continue
с      pause '8'
с72    format(1x,'степень-',i3,' коэффициент=',f36.18)

с-расчет модельных значений ym(i)
      k1=k+1
      izdelm(k)=1

```

```

kcdelm(k)=1
kdclm(k)=1
icdelm(k,1)=0
iddelm(k,1)=0
c delm(k)=0
do 54 i=1,n
  izym(i)=izak(km1)
  kcym(i)=kcak(km1)
  kdym(i)=kdak(km1)

      kcel=kcak(km1)
      kdel=kdak(km1)
      do 11051 l=1,kcel
        icym(i,l)=icak(km1,l)
11051 continue
      do 11052 l=1,kdel
        idym(i,l)=idak(km1,l)
11052 continue
c ym(i)=ak(km1)
do 540 j=1,m
  in=(j-1)*(k)+1
  ik=in+k-1
  isp=1
do 55 ist=in,ik
c ym(i)=ym(i)+ak(ist)*x(i,j)**(k-(isp-1))
  izw=1
  kcw=1
  kdw=1
  icw(1)=1
  idw(1)=0
  izr=izx(i,j)
  kcr=kcx(i,j)
  kdr=kdx(i,j)
do 11101 l=1,kcr
  icr(l)=icx(i,j,l)
11101 continue
do 11102 l=1,kdr
  idr(l)=idx(i,j,l)
11102 continue
  istep=k-(isp-1)
  if(istep.eq.0) goto 11103
do 11104 isw=1,istep
  call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
    *kcr,kdr,icr,idr,
    *kcrez,kdrez,icrez,idrez,krazr)
  izw=izrez
  kcw=kcrez
  kdw=kdrez
do 11105 l=1,kcw
  icw(l)=icrez(l)
11105 continue
do 11106 l=1,kdw
  idw(l)=idrez(l)
11106 continue
11104 continue
11103 continue
  izr=izak(ist)
  kcr=kcak(ist)
  kdr=kdak(ist)
do 11107 l=1,kcr
  icr(l)=icak(ist,l)
11107 continue

```

```

do 11108 l=1,kdr
idr(l)=idak(ist,l)
11108 continue
call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
izr=izym(i)
kcr=kcym(i)
kdr=kdym(i)
do 11109 l=1,kcr
icr(l)=icym(i,l)
11109 continue
do 11110 l=1,kdr
idr(l)=idym(i,l)
11110 continue
call slzn(izr,izrez,izr1,kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,
*kcl,kdl,icl,idl,krazr)
izym(i)=izr1
kcym(i)=kcl
kdym(i)=kdl
do 11111 l=1,kcl
icym(i,l)=icl(l)
11111 continue
do 11112 l=1,kdl
idym(i,l)=idl(l)
11112 continue
isp=isp+1
55 continue
540 continue
c pause '540'
c delm(k)=delm(k)+dabs(ym(i)-y(i))
izr=izym(i)
kcr=kcym(i)
kdr=kdym(i)
do 11201 l=1,kcr
icr(l)=icym(i,l)
11201 continue
do 11202 l=1,kdr
idr(l)=idym(i,l)
11202 continue
izr1=izy(i)
kcl=ky(i)
kdl=ky(i)
do 11301 l=1,kcl
icl(l)=icy(i,l)
11301 continue
do 11302 l=1,kdl
idl(l)=idy(i,l)
11302 continue
call wicz(izr,izr1,izrez,kcr,kdr,icr,idr,
*kcl,kdl,icl,idl,
*kcrez,kdrez,icrez,idrez,krazr)
izrez=izrez*izrez
izr=izdelm(k)
kcr=kcdelm(k)
kdr=kcddelm(k)
do 11401 l=1,kcr
icr(l)=icdelm(k,l)
11401 continue
do 11402 l=1,kdr
idr(l)=iddelm(k,l)
11402 continue

```

```

    call slzn(izr,izrez,izrl,kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,
*kcl,kdl,icl,idl,krazr)
    izdelm(k)=izrl
    kcdelm(k)=kcl
    kddelm(k)=kdl
    do 11501 l=1,kcl
    icdelm(k,l)=icl(l)
11501 continue

    do 11502 l=1,kdl
    iddelm(k,l)=idl(l)
11502 continue
345  format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
54   continue
c    pause '54'
c    delm(k)=delm(k)/n
    izr=izdelm(k)
    kcr=kcdelm(k)
    kdr=kddelm(k)
    do 11603 l=1,kcr
    icr(l)=icdelm(k,l)
11603 continue
    do 11604 l=1,kdr
    idr(l)=iddelm(k,l)
11604 continue
    call delzn(izr,izn,izrez,kcr,kdr,icr,idr,
*kcn,kdn,icn,idn,
*kcrez,kdrez,icrez,idrez,krazr)
    izdelm(k)=izrez
    kcdelm(k)=kcrez
    kddelm(k)=kdrez
    do 11701 l=1,kcrez
    icdelm(k,l)=icrez(l)
11701 continue
    do 11702 l=1,kdrez
    iddelm(k,l)=idrez(l)
11702 continue
    k=k+1
    if(k.le.kmax) goto 403
401  continue
c    pause '401'
    if(igau.eq.0.and.k.eq.1) goto 9988
    if(k.lt.kmax) kmax=k-1
    if(k.eq.kmax.and.igau.eq.0) kmax=k-1
c    write(5,*) 'вектор ошибок по степеням'
    do 4721 i=kna,kmax
c    write(5,4722) i,delm(i)
4722  format(1x,' степень=' ,i3,' ошибка=' ,d36.18)
4721  continue
    izxmi=izdelm(kna)
    kcxmi=kcdelm(kna)
    kdxmi=kddelm(kna)
    do 11721 l=1,kcxmi
    icxmi(l)=icdelm(kna,l)
11721 continue
    do 11722 l=1,kdxmi
    idxmi(l)=iddelm(kna,l)
11722 continue
c    xmin=delm(kna)
    imin=kna
    do 405 i=kna,kmax
    kcel=kcdelm(i)

```

```

kdel=kddelm(i)
if(kcel.gt.kcxmi) goto 405
if(kcel.lt.kcxmi) goto 12101
do 12103 l=1,kcel

    icr(l)=icdelm(i,l)
12103 continue
    do 12104 l=1,kdel
        idr(l)=iddelm(i,l)
12104 continue
    do 12102 l=1,kcel
        if(icr(l).gt.icxmi(l)) goto 405
        if(icr(l).lt.icxmi(l)) goto 12101
12102 continue
        kdro=kdel
        if(kdel.gt.kdxmi) kdro=kdxmi
        do 12601 l=1,kdro
            if(idr(l).gt.idxmi(l)) goto 405
            if(idr(l).lt.idxmi(l)) goto 12101
12601 continue
            if(kdel.ge.kdxmi) goto 405
c        if(kdel.eq.kdxmi) goto 405
c        if(delm(i).ge.xmin) goto 405
12101 continue
c        xmin=delm(i)
        izxmi=izdelm(i)
        kcxmi=kcdelm(i)
        kdxmi=kddelm(i)
        do 42601 l=1,kcxmi
            icxmi(l)=icdelm(i,l)
42601 continue

            do 12602 l=1,kdxmi
                idxmi(l)=iddelm(i,l)
12602 continue
            imin=i
405 continue
с-проверить необходимость метки 406
406 continue
с pause '406'
с-расчет модельных значений um(i) по min ошибке
write(6,701) imin
write(6,16532)
16532 format(/1x,'Модель для зависимого параметра'/)
kprod=7
iproduct=0
ipech=0
do 4001 iarg=1,karg
    narg=mnarg(iarg)
    write(6,4002) narg,(anaz(narg,jl),jl=1,60)
4002 format(/1x,'аргумент-',i3/1x,60a1)
    km1=imin*m+1
    km=imin*m
    kr=imin

    in=(iarg-1)*imin+1
    ik=in+imin-1
    do 876 j=in,ik
        write(6,12451) narg,kr
12451 format(1x,'номер аргумента-',i3,' степень=',i3/1x,
*' коэффициент')
        iz1=izakm(imin,j)
        kcel=kcakm(imin,j)

```

```

        kdel=kdakm(imin,j)
        do 12561 l=1,kcel
            icr(l)=icakm(imin,j,l)
12561 continue
            do 12562 l=1,kdel
                idr(l)=idakm(imin,j,l)
12562 continue
            write(6,12563) iz1
12563 format(1x,'знак=',i2)
            write(6,12564)
12564 format(1x,'целая часть коэффициента'//)

            write(6,12565) (icr(l),l=1,kcel)
12565 format(1x,70i1)
            write(6,12566)
12566 format(1x,'дробная часть коэффициента')

            write(6,12565) (idr(l),l=1,kdel)

c        write(9,50211) akm(imin,j),narg,kr
c        write(6,50211) akm(imin,j),narg,kr

        kr=kr-1
876 continue
4001 continue
c        write(6,14007) akm(imin,km1)
c        write(9,14007) akm(imin,km1)
c        write(*,14007) akm(imin,km1)
c        pause
            iz=izakm(imin,km1)
            kcel=kcakm(imin,km1)
            kdel=kdakm(imin,km1)
            do 13401 l=1,kcel
                icr(l)=icakm(imin,km1,l)
13401 continue
                do 13402 l=1,kdel
                    idr(l)=idakm(imin,km1,l)
13402 continue
                write(6,13403)
13403 format(/1x,'свободный член модели'//)

                write(6,13404) iz
13404 format(1x,'знак свободного члена=',i2)
                write(6,13405)
13405 format(1x,'целая часть свободного члена')
                write(6,12565) (icr(l),l=1,kcel)
                write(6,43405)
43405 format(1x,'дробная часть свободного члена')
                write(6,12565) (idr(l),l=1,kdel)
14007 format(5x,'+(',d23.15,')')
4007 format(5x,' +(',f25.6,')'//1x,
        *'*-умножение,**-возведение в степень'//)

c        delt=0
            izdelt=1
            kcdelt=1
            kddelt=1
            icdelt(1)=0
            iddelt(1)=0
            do 5201 i=1,m
                izwkl(i)=1
                kcwkl(i)=1

```

```

      kdwk1(i)=1
c      wk1(i)=0
      icwk1(i,1)=0
      idwk1(i,1)=0
5201      continue
      izsy=1
      kcsy=1
      kdsy=1
c      sy=0
      icsy(1)=0
      idsy(1)=0
      do 3354 i=1,n
c      ym(i)=akm(imin,kml)
      izym(i)=izakm(imin,kml)
      kcym(i)=kcakm(imin,kml)
      kdym(i)=kdakm(imin,kml)
      kcel=kcym(i)
      kdel=kdym(i)
      do 15601 l=1,kcel
15601      icym(i,l)=icakm(imin,kml,l)
      do 15602 l=1,kdel
15602      idym(i,l)=idakm(imin,kml,l)
      continue
      do 33540 j=1,m
      in=(j-1)*(imin)+1
      ik=in+imin-1
      isp=1
      do 3355 ist=in,ik
      istep=imin-(isp-1)
      izw=1
      kcw=1
      kdw=1
      icw(1)=1
      idw(1)=0
      izr=izx(i,j)
      kcr=kcx(i,j)
      kdr=kdx(i,j)
      do 31101 l=1,kcr
      icr(l)=icx(i,j,l)

31101      continue
      do 31102 l=1,kdr
      idr(l)=idx(i,j,l)
31102      continue
      if(istep.eq.0) goto 31103
      do 31104 isw=1,istep
      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
      izw=izrez
      kcw=kcrez
      kdw=kdrez
      do 31105 l=1,kcw
31105      icw(l)=icrez(l)
      do 31106 l=1,kdw
31106      idw(l)=idrez(l)
31106      continue
31104      continue
31103      continue
      izr=izakm(imin,ist)
      kcr=kcakm(imin,ist)

```

```

      kdr=kdakm(imin,ist)
      do 17601 l=1,kcr
        icr(l)=icakm(imin,ist,l)
17601 continue
      do 17602 l=1,kdr
        idr(l)=idakm(imin,ist,l)
17602 continue
      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
        *kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,krazr)
      iz1=izym(i)
      kcr=kcym(i)
      kdr=kdym(i)
      do 17301 l=1,kcr
        icr(l)=icym(i,l)
17301 continue
      do 17302 l=1,kdr
        idr(l)=idym(i,l)
17302 continue
      call slzn(iz1,izrez,iz2,kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,
        *kcl,kdl,icl,idl,krazr)
      izym(i)=iz2
      kcym(i)=kcl
      kdym(i)=kdl
      do 17303 l=1,kcl
        icym(i,l)=icl(l)
17303 continue
      do 17304 l=1,kdl
        idym(i,l)=idl(l)
17304 continue

c      ym(i)=ym(i)+akm(imin,ist)*x(i,j)**(imin-(isp-1))
      iz1=izwkl(j)
      kcel=kcwkl(j)
      kdel=kdwkl(j)
      do 18010 l=1,kcel
        icr(l)=icwkl(j,l)
18010 continue
      do 18011 l=1,kdel
        idr(l)=idwkl(j,l)
18011 continue
      izrez=izrez*izrez
      call slzn(iz1,izrez,iz2,kcel,kdel,icr,idr,
        *kcrez,kdrez,icrez,idrez,
        *kcl,kdl,icl,idl,krazr)
c      wkl(j)=wkl(j)+dabs(akm(imin,ist)*x(i,j)**(imin-(isp-1)))
      izwkl(j)=iz2
      kcwkl(j)=kcl
      kdwkl(j)=kdl
      do 18101 l=1,kcl
        icwkl(j,l)=icl(l)
18101 continue
      do 18102 l=1,kdl
        idwkl(j,l)=idl(l)
18102 continue

      isp=isp+1
3355 continue
33540 continue
      izr=izym(i)
      kcr=kcym(i)
      kdr=kdym(i)

```

```

do 19101 l=1,kcr
  icr(l)=icym(i,l)
19101 continue
do 19102 l=1,kdr
  idr(l)=idym(i,l)
19102 continue
  izl=izakm(imin,kml)
  kcl=kcakm(imin,kml)
  kdl=kdakm(imin,kml)
do 19103 l=1,kcl
  icl(l)=icakm(imin,kml,l)
19103 continue
do 19104 l=1,kdl
  idl(l)=idakm(imin,kml,l)
19104 continue
  call wiczn(izr,izl,izrez,kcr,kdr,icr,idr,
    *kcl,kdl,icl,idl,
    *kcrez,kdrez,icrez,idrez,krazr)
c   sy=sy+ym(i)-akm(imin,kml)
  call slzn(izsy,izrez,iz2,kcsy,kdsy,icsy,idsy,
    *kcrez,kdrez,icrez,idrez,
    *kcl,kdl,icl,idl,krazr)
  izsy=iz2
  kcsy=kcl
  kdsy=kdl
do 19201 l=1,kcl
  icsy(l)=icl(l)
19201 continue
do 19203 l=1,kdsy
  idsy(l)=idl(l)
19203 continue
  izr=izym(i)
  kcr=kcym(i)
  kdr=kdym(i)
do 19301 l=1,kcr
  icr(l)=icym(i,l)
19301 continue
do 19302 l=1,kdr
  idr(l)=idym(i,l)
19302 continue
c   write(6,*) i
c   write(6,*) 'модельное y'
c   write(6,*) izr
c   write(6,*) 'целая часть'
c   write(6,87321) (icr(l),l=1,kcr)
87321 format(1x,60i1)
c   write(6,*) 'дробная часть'
c   write(6,87321) (idr(l),l=1,kdr)
  izl=izy(i)
  kcl=кcy(i)
  kdl=кdy(i)
do 19401 l=1,kcl
  icl(l)=icy(i,l)
19401 continue
do 19402 l=1,kdl
  idl(l)=idy(i,l)
19402 continue
c   write(6,*) 'исходное y'
c   write(6,*) izl
c   write(6,*) 'целая часть'
c   write(6,87321) (icl(l),l=1,kcl)
c   write(6,*) 'дробная часть'
c   write(6,87321) (idl(l),l=1,kdl)

```

```

    call wiczn(izr,iz1,iz2,kcr,kdr,icr,idr,
    *kcl,kdl,icl,idl,
    *kcrez,kdrez,icrez,idrez,krazr)
    izrez=iz2*iz2
c    delt=delt+dabs(ym(i)-y(i))
    call slzn(izdelt,izrez,iz2,kcdelt,kddelt,icdelt,iddelt,
    *kcrez,kdrez,icrez,idrez,
    *kcl,kdl,icl,idl,krazr)
    izdelt=iz2
    kcdelt=kcl
    kddelt=kdl
    do 19501 l=1,kcl
    icdelt(l)=icl(l)
19501 continue
    do 19502 l=1,kdl
    iddelt(l)=idl(l)
19502 continue

c345 format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
3354 continue
    izwtot=1
    kcwtot=1
    kdwtot=1
    icwtot(1)=0
    idwtot(1)=0
c    wkltot=0
    do 5204 i=1,m
    izr=1
    kcr=kcwkl(i)
    kdr=kdwkl(i)
    do 19451 l=1,kcr
    icr(l)=icwkl(i,l)
19451 continue
    do 19452 l=1,kdr
    idr(l)=idwkl(i,l)
19452 continue
c    wkltot=wkltot+dabs(wkl(i))
    call slzn(izwtot,izr,iz1,kcwtot,kdwtot,icwtot,idwtot,
    *kcr,kdr,icr,idr,
    *kcrez,kdrez,icrez,idrez,krazr)
    izwtot=iz1
    kcwtot=kcrez
    kdwtot=kdrez
    do 19871 l=1,kcrez
    icwtot(l)=icrez(l)
19871 continue
    do 19872 l=1,kdrez
    idwtot(l)=idrez(l)
19872 continue
5204 continue
c    write(6,5307) sy,wkltot
5307 format(/1x,'sy=',f15.5,' wkltot=',f15.5/)
c    if(wkltot.eq.0) goto 5202
    do 5203 i=1,m
    izr=izwkl(i)
    kcr=kcwkl(i)
    kdr=kdwkl(i)
    do 19671 l=1,kcr
    icr(l)=icwkl(i,l)
19671 continue
    do 19672 l=1,kdr
    idr(l)=idwkl(i,l)

```

```

19672 continue
c   wkl(i)=wkl(i)/wkltot
    call delzn(izr,izwtot,izl,kcr,kdr,icr,idr,
*kcwtot,kdwtot,icwtot,idwtot,
*kcl,kdl,icl,idl,krazr)
    izwkl(i)=izl
    kcwkl(i)=kcl
    kdwkl(i)=kdl
    do 19345 l=1,kcl
      icwkl(i,l)=icl(l)
19345 continue
    do 19346 l=1,kdl
      idwkl(i,l)=idl(l)
19346 continue
5203 continue
5202 continue
c   delt=delt/n
    call delzn(izdelt,izn,izr,kcdelt,kddelt,icdelt,iddelt,
*kcn,kdn,icn,idn,
*kcrez,kdrez,icrez,idrez,krazr)
    izdelt=izr
    kcdelt=kcrez
    kddelt=kdrez
    do 19632 l=1,kcrez
      icdelt(l)=icrez(l)
19632 continue
    do 19633 l=1,kdrez
      iddelt(l)=idrez(l)
19633 continue

702  format(1x,6f13.5)
701  format(//1x,'оптимальная степень полинома=',i3//)
c   write(6,891) delt
    write(6,*) 'средняя абсолютная ошибка'
    write(6,*) 'целая часть ошибки'
    write(6,19711) (icdelt(l),l=1,kcdelt)
19711 format(1x,70i1)
    write(6,*) 'дробная часть ошибки'
    write(6,19711) (iddelt(l),l=1,kddelt)
c   write(6,5401)
5401 format(/18x,'Таблица-Вклады параметров-аргументов'/)
c   write(6,5402)
5402 format(11x,51('-'))
c   write(6,5403)
5403 format(11x,':Номер:',6x,'Название параметра',6x,':',3x,
*'Вклад',4x,':')
c   write(6,5402)
    prob=' '
    do 5404 i=1,m
      narg=mnarg(i)
      write(6,5405) narg,(anaz(narg,j),j=1,60)
5405 format(1x,'номер аргумента=',i3/1x,60a1)
      write(6,*) 'целая часть вклада'
      kcel=kcwkl(i)
      write(6,19711) (icwkl(i,l),l=1,kcel)
      write(6,*) 'дробная часть вклада'
      kdel=kdwkl(i)
      write(6,19711) (idwkl(i,l),l=1,kdel)

5404 continue
891  format(/1x,'средняя абсолютная ошибка=',f36.18/)
9988 continue

```

```

2001 continue
      close(4,status='keep')
      close(6,status='keep')
      close(5,status='keep')
      close(7,status='keep')
      end
$large
      subroutine mnk(n,m,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
      *izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,igau,krazr)
с-метод наименьших квадратов
с      dimension x(100,20),y(100),ak(100),ym(100),c(100,100),f(100)
с      dimension weko(100)
с      double precision x,y,ak,ym,c,f,weko,sum
      integer*2 izf(31),kcf(31),kdf(31),icf(31,20),idf(31,20)
      integer*2 icw(400),idw(400),izx(50,10),kcx(50,10),kdx(50,10)
      integer*2 icx(50,10,20),idx(50,10,20),icrez(400),idrez(400),
      *izy(50),kcy(50),kdy(50),icy(50,20),idy(50,20),icr(400),idr(400),
      *izc(31,31),kcc(31,31),kdc(31,31),icc(31,31,20),idc(31,31,20),
      *icw1(400),idw1(400),icl(400),idl(400),icn(400),idn(400)
      integer*2 izak(31),kcak(31),kdak(31),icak(31,20),idak(31,20)

      integer s,p
с-с(50,50),матрица системы нормальных уравнений
с-f(50),вектор свободных членов
с-ak(50),коэффициенты полиномов по аргументам
с-x(50,10),матрица аргументов
с-y(50),массив значений функции
с-ym(50),массив модельных значений функции
с      write(6,*) '          Метод наименьших квадратов'
с      do 1 i=1,n

с      write(6,123) i,x(i),i,y(i)
123  format(1x,'x(',i3,')=',f13.5,' y(',i3,')=',f13.5)
1    continue
с-определение матрицы с
с-i-номер наблюдения,строки
с-j-номер параметра-аргумента
с-l-номер коэффициента,степени
с-s-номер степени,p-номер параметра
с-m-количество параметров-аргументов
с-k,порядок полиномов
с-n,число значений функции
с      write(6,*) 'передача в mnk'
с      write(6,*) 'матрица x'
с      do 34521 i=1,n
с      do 34522 j=1,m

с      write(6,*) i,j
с      write(6,*) 'целая часть x'
с      izn=izx(i,j)
с      write(6,*) izn
с      kcel=kcx(i,j)
с      kdel=kdx(i,j)
с      write(6,34523) (icx(i,j,l),l=1,kcel)
34523 format(1x,50i1)
с      write(6,*) 'дробная часть x'
с      write(6,34523) (idx(i,j,l),l=1,kdel)
34522 continue
34521 continue
с      write(6,*) 'вектор y'
с      do 34545 i=1,n
с      kcel=kcy(i)
с      kdel=kdy(i)

```

```

c      write(6,*) izy(i)
c      write(6,*) 'целая часть y'
c      write(6,34523) (icy(i,1),l=1,kcel)
c      write(6,*) 'дробная часть y '
c      write(6,34523) (idy(i,1),l=1,kdel)
34545 continue
c      stop
c      k1=k+1
c      pause 'mnk1'
c      do 2 p=1,m
c      do 282 s=1,k
c      izf((p-1)*k+s)=1
c      kcf((p-1)*k+s)=1
c      kdf((p-1)*k+s)=1
c      icf((p-1)*k+s,1)=0
c      idf((p-1)*k+s,1)=0
c      f((p-1)*k+s)=0
c      do 2002 i=1,n
c      izw=1
c      kcw=1
c      kdw=1
c      icw(1)=1
c      idw(1)=0
c      izr=izx(i,p)
c      kcr=kcx(i,p)
c      kdr=kdx(i,p)
c      do 12101 l=1,kcr
c      icr(l)=icx(i,p,l)
12101 continue
c      do 12102 l=1,kdr
c      idr(l)=idx(i,p,l)
12102 continue
c      istep=k-(s-1)
c      if(istep.eq.0) goto 32103
c      do 12104 l=1,istep
c      pause '1'
c      write(6,*) 'p1'
c      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
c      *kcr,kdr,icr,idr,
c      *kcrez,kdrez,icrez,idrez,krazr)
c      write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez
c      pause '2'
c      izw=izrez
c      kcw=kcrez
c      kdw=kdrez
c      do 12201 lp=1,kcrez
c      icw(lp)=icrez(lp)
12201 continue
c      do 12202 lp=1,kdrez
c      idw(lp)=idrez(lp)
12202 continue
12104 continue
32103 continue
c      pause '32103'
c      izr=izy(i)
c      kcr=kcy(i)
c      kdr=kdy(i)
c      do 13291 l=1,kcr
c      icr(l)=icy(i,l)
13291 continue
c      do 13282 l=1,kdr
c      idr(l)=idy(i,l)
13282 continue

```

```

c   pause '3'
c   write(6,*) 'p3'
c   call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcw,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
c   write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez
c   pause '4'
c   ilf=(p-1)*k+s
c   izr=izf(ilf)
c   kcr=kcf(ilf)
c   kdr=kdf(ilf)
c   do 14301 l=1,kcr
c   icr(l)=icf(ilf,l)
c   write(*,*) icr(l)
14301 continue
c   do 14302 l=1,kdr
c   idr(l)=idf(ilf,l)
c   write(*,*) idr(l)
14302 continue
c   pause '5 na'
c   write(6,*) 'первое слагаемое'
c   write(6,*) izr,kcr,kdr
c   write(6,*) izr,kcr,kdr
c   pause '5_1'
c   write(6,*) izr
c   write(6,49701) izr
49701 format(1x,'знак=',i3/)
c   write(6,*) 'целая часть'
c   write(6,*) (icr(l),l=1,kcr)
c   pause '5_1_1'
c   write(6,*) (icr(l),l=1,kcr)
c   pause
49702 format(1x,50i1)
c   write(6,*) 'дробная часть'
c   write(6,*) (idr(l),l=1,kdr)
49703 format(1x,50i1)
c   write(6,*) 'второе слагаемое'
c   write(6,49701) izrez
c   write(6,*) izrez,kcrez,kdrez
c   pause '5_2'
c   write(6,*) 'целая часть'
c   write(6,*) (icrez(l),l=1,kcrez)
c   write(6,*) 'дробная часть'
c   write(6,*) (idrez(l),l=1,kdrez)
c   pause '5'
c   call slzn(izr,izrez,izl,kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,
*kcl,kdl,icl,idl,krazr)
c   pause '6'
c   f((p-1)*k+s)= f((p-1)*k+s)+y(i)*x(i,p)**(k-(s-1))
c   izf(ilf)=izl
c   kcf(ilf)=kcl
c   kdf(ilf)=kdl
c   do 15401 l=1,kcl
c   icf(ilf,l)=icl(l)
15401 continue
c   do 15402 l=1,kdl
c   idf(ilf,l)=idl(l)
15402 continue
2002 continue
c   pause '2002'
c   do 212 j=1,m
c   do 211 l=1,k

```

```

      ilc=(p-1)*k+s
      jlc=(j-1)*k+1
c      c((p-1)*k+s,(j-1)*k+1)=0
      izc(ilc,jlc)=1
      kcc(ilc,jlc)=1
      kdc(ilc,jlc)=1
      icc(ilc,jlc,1)=0
      idc(ilc,jlc,1)=0
      do 3 i=1,n
c      c((p-1)*k+s,(j-1)*k+1)=c((p-1)*k+s,(j-1)*k+1)+
c      *x(i,j)**(k-(l-1))*x(i,p)**(k-(s-1))
      izw=1
      kcw=1
      kdw=1
      icw(1)=1
      idw(1)=0
      istep=k-(l-1)
      izr=izx(i,j)
      kcr=kcx(i,j)
      kdr=kdx(i,j)
      do 14201 lp=1,kcr
14201 icr(lp)=icx(i,j,lp)
      continue
      do 14202 lp=1,kdr
      idr(lp)=idx(i,j,lp)
14202 continue
      if(istep.eq.0) goto 16101
      do 16102 lpr=1,istep
c      pause '7'
      call ymnzn(izw,izr,izl,kcw,kdw,icw,idw,
*c      *kcr,kdr,icr,idr,
*c      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '8'
      izw=izl
      kcw=kcrez
      kdw=kdrez
      do 17101 lp=1,kcw
17101 icw(lp)=icrez(lp)
      continue
      do 17102 lp=1,kdw
      idw(lp)=idrez(lp)
17102 continue
16102 continue
16101 continue
      istep=k-(s-1)
      izw1=1
      kcw1=1
      kdw1=1
      icw1(1)=1
      idw1(1)=0
      izr=izx(i,p)
      kcr=kcx(i,p)
      kdr=kdx(i,p)
      do 18101 lp=1,kcr
18101 icr(lp)=icx(i,p,lp)
      continue
      do 18102 lp=1,kdr
      idr(lp)=idx(i,p,lp)
18102 continue
      if(istep.eq.0) goto 18301
      do 18302 lpr=1,istep
c      pause '9'
      call ymnzn(izw1,izr,izrez,kcw1,kdw1,icw1,idw1,

```

```

      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '10'
      izw1=izrez
      kcw1=kcrez
      kdwl=kdrez
      do 13201 lp=1,kcrez
      icw1(lp)=icrez(lp)
13201 continue
      do 63201 lp=1,kdrez
      idw1(lp)=idrez(lp)
63201 continue
18302 continue
18301 continue
c      pause '11'
      call ymnzn(izw,izw1,izrez,kcw,kdw,icw,idw,
      *kcw1,kdw1,icw1,idw1,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '12'
      izr=izc(ilc,jlc)
      kcr=kcc(ilc,jlc)
      kdr=kdc(ilc,jlc)
      do 17701 lp=1,kcr
      icr(lp)=icc(ilc,jlc,lp)
17701 continue
      do 17702 lp=1,kdr
      idr(lp)=idc(ilc,jlc,lp)
17702 continue
c      pause '13'
      call slzn(izrez,izr,izl,kcrez,kdrez,icrez,idrez,
      *kcr,kdr,icr,idr,
      *kcl,kdl,icl,idl,krazr)
c      pause '14'
      izc(ilc,jlc)=izl
      kcc(ilc,jlc)=kcl
      kdc(ilc,jlc)=kdl
      do 16701 lp=1,kcl
      icc(ilc,jlc,lp)=icl(lp)
16701 continue
      do 16702 lp=1,kdl
      idc(ilc,jlc,lp)=idl(lp)
16702 continue
3      continue
      istr=(p-1)*k+s
      isto=(j-1)*k+l
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
3311 format(1x,'p=',i3,' s=',i3,' j=',i3,' l=',i3/1x,
*'istr=',i3,' isto=',i3,' c=',f15.5,' f=',f15.5/)
211 continue
212 continue
      ilc=(p-1)*k+s
      jlc=(m-1)*k+k+1
      izc(ilc,jlc)=1
      kcc(ilc,jlc)=1
      kdc(ilc,jlc)=1
      icc(ilc,jlc,1)=0
      idc(ilc,jlc,1)=0
c      c((p-1)*k+s,(m-1)*k+k+1)=0
      do 703 i=1,n
      istep=k-(s-1)
      izw=1
      kcw=1
      kdwl=1

```

```

        icw(1)=1
        idw(1)=0
        izr=izx(i,p)
        kcr=kcx(i,p)
        kdr=kdx(i,p)
        do 18401 lp=1,kcr
18401 icr(lp)=icx(i,p,lp)
        continue
        do 18402 lp=1,kdr
18402 idr(lp)=idx(i,p,lp)
        continue
        if(istep.eq.0) goto 18403
        do 18404 lpr=1,istep
c      pause '15'
        call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
        *kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,krazr)
c      pause '16'
        izw=izrez
        kw=kcrez
        kdw=kdrez
        do 16785 lp=1,kcrez
16785 icw(lp)=icrez(lp)
        continue
        do 16786 lp=1,kdrez
16786 idw(lp)=idrez(lp)
        continue
18404 continue
18403 continue
        izr=izc(ilc,jlc)
        kcr=kcc(ilc,jlc)
        kdr=kdc(ilc,jlc)
        do 11301 lp=1,kcr
11301 icr(lp)=icc(ilc,jlc,lp)
        continue
        do 11302 lp=1,kdr
11302 idr(lp)=idc(ilc,jlc,lp)
        continue
c      pause '17'
        call slzn(izw,izr,izrez,kcw,kdw,icw,idw,
        *kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,krazr)
c      pause '18'
        izc(ilc,jlc)=izrez
        kcc(ilc,jlc)=kcrez
        kdc(ilc,jlc)=kdrez
        do 13781 lp=1,kcrez
13781 icc(ilc,jlc,lp)=icrez(lp)
        continue
        do 13782 lp=1,kdrez
13782 idc(ilc,jlc,lp)=idrez(lp)
        continue

c      c((p-1)*k+s,(m-1)*k+k+1)=c((p-1)*k+s,(m-1)*k+k+1)+
c      *x(i,p)**(k-(s-1))

703  continue
        istr=(p-1)*k+s
        isto=(m-1)*k+k+1
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
282  continue
2      continue

```

```

inf=(m-1)*k+k+1
izf (inf)=1
kcf (inf)=1
kdf (inf)=1
icf (inf,1)=0
idf (inf,1)=0
c f((m-1)*k+k+1)=0
do 12002 i=1,n
  izr=izf (inf)
  kcr=kcf (inf)
  kdr=kdf (inf)
  do 14781 lp=1,kcr
    icr (lp)=icf (inf,lp)
14781 continue
  do 14782 lp=1,kdr
    idr (lp)=idf (inf,lp)
14782 continue
  iz1=izy (i)
  kcl=kcy (i)
  kdl=kdy (i)
  do 14785 lp=1,kcl
    icl (lp)=icy (i,lp)
14785 continue
  do 14786 lp=1,kdl
    idl (lp)=idy (i,lp)
14786 continue
c pause '19'
  call slzn (izr,iz1,izrez,kcr,kdr,icr,idr,
    *kcl,kdl,icl,idl,
    *kcrez,kdrez,icrez,idrez,krazr)
c pause '20'
  izf (inf)=izrez
  kcf (inf)=kcrez
  kdf (inf)=kdrez
  do 44871 lp=1,kcrez
    icf (inf,lp)=icrez (lp)
44871 continue
  do 14872 lp=1,kdrez
    idf (inf,lp)=idrez (lp)
14872 continue
c f((m-1)*k+k+1)= f((m-1)*k+k+1)+y(i)
12002 continue
  do 4212 j=1,m
    do 4211 l=1,k
      icc1=(m-1)*k+k+1
      jc1=(j-1)*k+1
      izc (icc1,jc1)=1
      kcc (icc1,jc1)=1
      kdc (icc1,jc1)=1
      icc (icc1,jc1,1)=0
      idc (icc1,jc1,1)=0
c c((m-1)*k+k+1,(j-1)*k+1)=0
      do 371 i=1,n
        istep=k-(l-1)
        izw=1
        kcw=1
        kdw=1
        icw(1)=1
        idw(1)=0
        izr=izx (i,j)
        kcr=kcx (i,j)
        kdr=kdx (i,j)
        do 23671 lp=1,kcr

```

```

        icr(lp)=icx(i,j,lp)
23671 continue
        do 23672 lp=1,kdr
            idr(lp)=idx(i,j,lp)
23672 continue
            if(istep.eq.0) goto 23673
            do 23674 lpr=1,istep
c           pause '21'
            call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
                *kcr,kdr,icr,idr,
                *kcrez,kdrez,icrez,idrez,krazr)
c           pause '22'
            izw=izrez
            kw=kcrez
            kdw=kdrez
            do 24301 lp=1,kcrez
                icw(lp)=icrez(lp)
24301 continue
            do 24302 lp=1,kdrez
                idw(lp)=idrez(lp)
24302 continue
23674 continue
23673 continue
            izr=izc(iccl,jcl)
            kcr=kcc(iccl,jcl)
            kdr=kdc(iccl,jcl)
            do 25401 lp=1,kcr
                icr(lp)=icc(iccl,jcl,lp)
25401 continue
            do 25402 lp=1,kdr
                idr(lp)=idc(iccl,jcl,lp)
25402 continue
c           pause '23'
            call slzn(izw,izr,izrez,kcw,kdw,icw,idw,
                *kcr,kdr,icr,idr,
                *kcrez,kdrez,icrez,idrez,krazr)
c           pause '24'
            izc(iccl,jcl)=izrez
            kcc(iccl,jcl)=kcrez
            kdc(iccl,jcl)=kdrez
            do 26701 lp=1,kcrez
                icc(iccl,jcl,lp)=icrez(lp)
26701 continue
            do 26702 lp=1,kdrez
                idc(iccl,jcl,lp)=idrez(lp)
26702 continue
c           c((m-1)*k+k+1,(j-1)*k+1)=c((m-1)*k+k+1,(j-1)*k+1)+
c           *x(i,j)**(k-(l-1))
371 continue
            istr=(m-1)*k+k+1
            isto=(j-1)*k+1
c           write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
c3311         format(1x,'p=',i3,' s=',i3,' j=',i3,' l=',i3/1x,
c           *' istr=',i3,' isto=',i3,' c=',f15.5,' f=',f15.5/)
4211 continue
4212 continue
            iccl=(m-1)*k+k+1
            izc(iccl,iccl)=izn
            kcc(iccl,iccl)=kcn
            kdc(iccl,iccl)=kdn
            do 27101 lp=1,kcn
                icc(iccl,iccl,lp)=icn(lp)
27101 continue

```

```

do 27102 lp=1,kdn
  idc(icc1,icc1,lp)=idn(lp)
27102 continue
c      c((m-1)*k+k+1,(m-1)*k+k+1)=n

  istr=(m-1)*k+k+1
  isto=(m-1)*k+k+1
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)

  kraz=m*k+1
c      write(5,*) ' матрица c'
      open(9,file='promprow',status='Unknown')
c      write(5,*) kraz
      write(9,*) kraz,krazr
      do 56 i=1,kraz
      do 1234 j=1,kraz
c      write(5,4581) i,j
4581 format(1x,'ic=',i3,' jc=',i3)
c      write(5,*) 'целая часть'
      izr=izc(i,j)
      kcel=kcc(i,j)
      kdel=kdc(i,j)

      write(9,33357) izr,kcel,kdel,(icc(i,j,jp),jp=1,kcel),
* (idc(i,j,jp),jp=1,kdel)
33357 format(3i2,80i1)
1234 continue
56 continue
57 format(1x,70i1)
c      write(6,*) ' вектор свободных членов'
c4981 continue
      do 56701 i=1,kraz
      izr=izf(i)
      kcel=kcf(i)
      kdel=kdf(i)
c      write(6,56702) i
56702 format(1x,'fi=',i6)
c      write(6,*) 'целая часть'

      write(9,33357) izr,kcel,kdel,(icf(i,ip),ip=1,kcel),
* (idf(i,ip),ip=1,kdel)
56701 continue
      close(9,status='keep')
      kg=k*m+1
c      call gay(kg,c,f,ak,igau)
c      pause 'pered gaydl'
      call gaydl(kg,izc,kcc,kdc,icc,idc,izf,kcf,kdf,icf,idf,
*izak,kcak,kdak,icak,idak,igau,krazr)
c      pause 'posle gaydl'
      kg=k*m+1
      call prowg(kg,izc,kcc,kdc,icc,idc,
*izf,kcf,kdf,icf,idf,izak,kcak,kdak,icak,idak,igau,krazr)

      if(igau.eq.0) return
      return
      end

```

## Набор dlmnfraz.for(поразрядный ввод целой и дробной частей)

\$large

c-метод наименьших квадратов в длинной арифметике

```

c      dimension x(100,20),y(100),akm(10,100),mnarg(20),xr(100)
c      dimension ym(100),ak(100),delm(10),zgl(80),rab(100,150)
c      dimension anaz(150,60),wkl(20)
c      double precision ym,dx,x,y,ak,akm,delm,delt,rab,xr,wkl,wkltot,sy
c      double precision xmin
c      dimension anaz(20,60),mnarg(20),zgl(80)
c      integer*2 izrab(50,20),kcrab(50,20),kdrab(50,20),
*icrab(50,20,20),idrab(50,20,20),izy(50),kcy(50),kdy(50),
*icy(50,20),idy(50,20),izx(50,10),kcx(50,10),kdx(50,10),
*icx(50,10,20),idx(50,10,20),izakm(3,31),izak(31),kcakm(3,31),
*kcak(31),kdakm(3,31),kdak(31),icakm(3,31,20),icak(31,20),
*idakm(3,31,20),idak(31,20)
c      integer*2 izdelm(3),kcdelm(3),kddelm(3),icdelm(3,20),
*iddelm(3,20),izym(50),kcym(50),kdy(50),icy(50,20),idy(50,20),
*icw(400),idw(400),icr(400),idr(400),icrez(400),idrez(400),
*icl(400),
*idl(400),icn(400),idn(400),icxmi(400),idxmi(400),icdelt(400),
*iddelt(400),izwkl(10),kcwkl(10),kdwkl(10),icwkl(10,20),
*idwkl(10,20)
c      integer*2 icsy(400),idsy(400),icwtot(400),idwtot(400)
c      double precision rwwod(20)
c      open(4,file='sledraz',status='old')
c      open(6,file='dlmnfraz',status='Unknown')
c      open(5,file='prowfraz',status='Unknown')
c      open(7,file='name',status='old')
c      open(9,file='prompr2',status='Unknown')
c      read(7,*) knaz
c      if(knaz.gt.20) goto 1762
c      do 1441 i=1,knaz
c      do 83 j=1,60
c      anaz(i,j)= ' '
83      continue
c      read(7,1442) (anaz(i,j),j=1,60)
1442 format(60a1)
1441 continue

c      read(4,1001) zgl
1001 format(80a1)
c      write(6,1001) zgl
c      read(4,*) n,m
c      read(4,*) karg
c      read(4,*) (mnarg(i),i=1,karg)
c      read(4,*) inay,ikoу
c      read(4,*) kmax
c      read(4,*) krazr,ka
c      if(n.le.50) goto 1443
c      write(6,*) 'число строчек-наблюдений должно быть<=50'
c      stop
1443 continue
c      if(m.le.20) goto 1444
1762 continue
c      write(6,*) 'число параметров-столбиков должно быть<=20'
c      stop
1444 continue
c      nt=n
c      l=0
11601 continue
c      ios=nt-nt/10*10

```

```

l=l+1
idl(l)=ios
nt=nt/10
if(nt.gt.0) goto 11601
izn=1
kcn=1
kdn=1
lt=1
do 11602 l=1,kcn
icn(l)=idl(lt)
lt=lt-1
11602 continue
idn(l)=0

с-ввод исходной матрицы в длинном формате
do 1002 i=1,n
do 11001 j=1,m
write(6,56721) i,j
56721 format(/1x,'i=',i3,' j=',i3/)
read(4,*) izrab(i,j),kcrab(i,j),kdrab(i,j)
write(6,56722) izrab(i,j)
56722 format(1x,' знак=',i2/)
kcel=kcrab(i,j)
kdel=kdrab(i,j)
read(4,*) (icrab(i,j,jl),jl=1,kcel)
read(4,*) (idrab(i,j,jl),jl=1,kdel)
write(6,*) 'целая часть'
write(6,19781) (icrab(i,j,l),l=1,kcel)
19781 format(1x,50i1)
write(6,*) 'дробная часть'
write(6,19781) (idrab(i,j,l),l=1,kdel)
11002 format(70i1)
11001 continue
с read(4,*) (rab(i,j),j=1,m)
с write(6,*) (rab(i,j),j=1,m)
1002 continue
7775 continue
с stop
с write(*,*) 'введите максимальную степень<=10'
с read(*,*) kmax
с write(*,*) 'введите начальную степень'
с read(*,*) kna
кna=1
к=kna
с write(*,*) 'Введите количество параметров-аргументов<=10'
с read(*,*) karg
с kmax=100/karg-1
с if(kmax.gt.10) kmax=6
с kmax=4
с write(*,*) 'введите через пробел номера аргументов'
с read(*,*) (mnarg(i),i=1,karg)
с write(*,1003)
1003 format(1x,'Введите через пробел начальный и конечный номер'/1x,
*'зависимой переменной')
с read(*,*) inay,ikoу
с write(*,*) 'введите степень полинома<=3'
с read(*,*) kmax
с write(*,*) 'введите количество разрядов для арифметики<=20'
с read(*,*) krazr
write(6,13667) kmax
13667 format(/1x,'степень полинома=',i2//)
m=karg

```

```

do 2001 ipy=inay,ikoу
k=kna
write(6,2003) ipy,(anaz(ipy,jl),jl=1,60)
write(6,*) 'номера аргументов'
write(6,2004) (mnarg(i),i=1,karg)
403 continue
с-формирование вектора у для длинной арифметики
do 2002 i=1,n
izy(i)=izrab(i,ipy)
kcy(i)=kcrab(i,ipy)
kdy(i)=kdrab(i,ipy)
kcel=kcy(i)
kdel=kdy(i)

do 11021 l=1,kcel
icy(i,l)=icrab(i,ipy,l)
11021 continue
do 11022 l=1,kdel
idy(i,l)=idrab(i,ipy,l)
11022 continue
с y(i)=rab(i,ipy)
2002 continue
2003 format(1x,'зависимый параметр-',i3/1x,60a1/)
2004 format(1x,15i5)
с write(6,*) 'исходный зависимый вектор'
с write(6,*) (y(i),i=1,n)
с-формирование матрицы х аргументов для длинной арифметики
с pause 'х'
ila=0
do 9001 iarg=1,karg
narg=mnarg(iarg)
с write(6,1231) narg
1231 format(1x,'arg=',i3)
ila=ila+1
do 9002 i=1,n
izx(i,ila)=izrab(i,narg)
kcx(i,ila)=kcrab(i,narg)
kdx(i,ila)=kdrab(i,narg)
kcel=kcx(i,ila)
kdel=kdx(i,ila)
do 11031 l=1,kcel
icx(i,ila,l)=icrab(i,narg,l)
11031 continue
do 11032 l=1,kdel
idx(i,ila,l)=idrab(i,narg,l)
11032 continue
с x(i,ila)=rab(i,narg)
9002 continue
9001 continue
с write(6,*) 'matrix x'
do 9004 i=1,n
с write(6,8771) (x(i,j),j=1,karg)
8771 format(1x,5f13.4)
9004 continue
с write(6,*) 'зависимый вектор'
с write(6,8771) (y(i),i=1,n)
с pause 'pered mnk'
call mnk(n,karg,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
*izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,igau,krazr)
с pause 'posle mnk'
с write(*,*) igau
if(igau.eq.0) goto 401

```

```

      kml=k*m+1
c      write(6,123) k
c      write(6,124) (ak(ip),ip=1,kml)
124    format(1x,6f13.5)
123    format(1x,'k=',i3/1x,6f12.5)
      do 8 i=1,kml
        izakm(k,i)=izak(i)
        kcakm(k,i)=kcak(i)
        kdakm(k,i)=kdak(i)
        kcel=kcak(i)
        kdel=kdak(i)
        do 11041 l=1,kcel
          icakm(k,i,l)=icak(i,l)
11041 continue
        do 11042 l=1,kdel
          idakm(k,i,l)=idak(i,l)
11042 continue
c      akm(k,i)=ak(i)
8      continue
c      pause '8'
c72    format(1x,'степень-',i3,' коэффициент=',f36.18)

```

с-расчет модельных значений  $y_m(i)$

```

      k1=k+1
      izdelm(k)=1
      kcdelm(k)=1
      kddelm(k)=1
      icdelm(k,1)=0
      iddelm(k,1)=0
c      delm(k)=0
      do 54 i=1,n
        izym(i)=izak(kml)
        kcy(m,i)=kcak(kml)
        kdym(i)=kdak(kml)

        kcel=kcak(kml)
        kdel=kdak(kml)
        do 11051 l=1,kcel
          icy(m,i,l)=icak(kml,l)
11051 continue
        do 11052 l=1,kdel
          idym(i,l)=idak(kml,l)
11052 continue
c      ym(i)=ak(kml)
        do 540 j=1,m
          in=(j-1)*(k)+1
          ik=in+k-1
          isp=1
          do 55 ist=in,ik
c      ym(i)=ym(i)+ak(ist)*x(i,j)**(k-(isp-1))
          izw=1
          kcw=1
          kdw=1
          icw(1)=1
          idw(1)=0
          izr=izx(i,j)
          kcr=kcx(i,j)
          kdr=kdx(i,j)
          do 11101 l=1,kcr
            icr(l)=icx(i,j,l)
11101 continue
          do 11102 l=1,kdr

```

```

        idr(l)=idx(i,j,l)
11102 continue
        istep=k-(isp-1)
        if(istep.eq.0) goto 11103
        do 11104 isw=1,istep
        call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
        *kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,krazr)
        izw=izrez
        kcw=kcrez
        kdw=kdrez
        do 11105 l=1,kcw
        icw(l)=icrez(l)
11105 continue
        do 11106 l=1,kdw
        idw(l)=idrez(l)
11106 continue
11104 continue
11103 continue
        izr=izak(ist)
        kcr=kcak(ist)
        kdr=kdak(ist)
        do 11107 l=1,kcr
        icr(l)=icak(ist,l)
11107 continue
        do 11108 l=1,kdr
        idr(l)=idak(ist,l)
11108 continue
        call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
        *kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,krazr)
        izr=izym(i)
        kcr=kcym(i)
        kdr=kdym(i)
        do 11109 l=1,kcr
        icr(l)=icym(i,l)
11109 continue
        do 11110 l=1,kdr
        idr(l)=idym(i,l)
11110 continue
        call slzn(izr,izrez,izr1,kcr,kdr,icr,idr,
        *kcrez,kdrez,icrez,idrez,
        *kcl,kdl,icl,idl,krazr)
        izym(i)=izr1
        kcym(i)=kcl
        kdym(i)=kdl
        do 11111 l=1,kcl
        icym(i,l)=icl(l)
11111 continue
        do 11112 l=1,kdl
        idym(i,l)=idl(l)
11112 continue
        isp=isp+1
55 continue
540 continue
c pause '540'
c delm(k)=delm(k)+dabs(ym(i)-y(i))
        izr=izym(i)
        kcr=kcym(i)
        kdr=kdym(i)
        do 11201 l=1,kcr
        icr(l)=icym(i,l)
11201 continue

```

```

do 11202 l=1,kdr
idr(l)=idy(i,l)
11202 continue
izr1=izy(i)
kcl=kcy(i)
kdl=kdy(i)
do 11301 l=1,kcl
icl(l)=icy(i,l)
11301 continue
do 11302 l=1,kdl
idl(l)=idy(i,l)
11302 continue
call wiczn(izr,izr1,izrez,kcr,kdr,icr,idr,
*kcl,kdl,icl,idl,
*kcrez,kdrez,icrez,idrez,krazr)
izrez=izrez*izrez
izr=izdelm(k)
kcr=kcdelm(k)
kdr=kddelm(k)
do 11401 l=1,kcr
icr(l)=icdelm(k,l)
11401 continue
do 11402 l=1,kdr
idr(l)=iddelm(k,l)
11402 continue
call slzn(izr,izrez,izr1,kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,
*kcl,kdl,icl,idl,krazr)
izdelm(k)=izr1
kcdelm(k)=kcl
kddelm(k)=kdl
do 11501 l=1,kcl
icdelm(k,l)=icl(l)
11501 continue

do 11502 l=1,kdl
iddelm(k,l)=idl(l)
11502 continue
345 format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
54 continue
c pause '54'
c delm(k)=delm(k)/n
izr=izdelm(k)
kcr=kcdelm(k)
kdr=kddelm(k)
do 11603 l=1,kcr
icr(l)=icdelm(k,l)
11603 continue
do 11604 l=1,kdr
idr(l)=iddelm(k,l)
11604 continue
call delzn(izr,izn,izrez,kcr,kdr,icr,idr,
*kcn,kdn,icn,idn,
*kcrez,kdrez,icrez,idrez,krazr)
izdelm(k)=izrez
kcdelm(k)=kcrez
kddelm(k)=kdrez
do 11701 l=1,kcrez
icdelm(k,l)=icrez(l)
11701 continue
do 11702 l=1,kdrez
iddelm(k,l)=idrez(l)
11702 continue

```

```

        k=k+1
        if(k.le.kmax) goto 403
401      continue
c       pause '401'
        if(igau.eq.0.and.k.eq.1) goto 9988
        if(k.lt.kmax) kmax=k-1
        if(k.eq.kmax.and.igau.eq.0) kmax=k-1
c       write(5,*) 'вектор ошибок по степеням'
        do 4721 i=kna,kmax
c       write(5,4722) i,delm(i)
4722    format(1x,'степень=',i3,' ошибка=',d36.18)
4721    continue
        izxmi=izdelm(kna)
        kcxmi=kcdelm(kna)
        kdxmi=kddelm(kna)
        do 11721 l=1,kcxmi
        icxmi(l)=icdelm(kna,l)
11721    continue
        do 11722 l=1,kdxmi
        idxmi(l)=iddelm(kna,l)
11722    continue
c       xmin=delm(kna)
        imin=kna
        do 405 i=kna,kmax
        kcel=kcdelm(i)
        kdel=kddelm(i)
        if(kcel.gt.kcxmi) goto 405
        if(kcel.lt.kcxmi) goto 12101
        do 12103 l=1,kcel
        icr(l)=icdelm(i,l)
12103    continue
        do 12104 l=1,kdel
        idr(l)=iddelm(i,l)
12104    continue
        do 12102 l=1,kcel
        if(icr(l).gt.icxmi(l)) goto 405
        if(icr(l).lt.icxmi(l)) goto 12101
12102    continue
        kdro=kdel
        if(kdel.gt.kdxmi) kdro=kdxmi
        do 12601 l=1,kdro
        if(idr(l).gt.idxmi(l)) goto 405
        if(idr(l).lt.idxmi(l)) goto 12101
12601    continue
        if(kdel.ge.kdxmi) goto 405
c       if(kdel.eq.kdxmi) goto 405
c       if(delm(i).ge.xmin) goto 405
12101    continue
c       xmin=delm(i)
        izxmi=izdelm(i)
        kcxmi=kcdelm(i)
        kdxmi=kddelm(i)
        do 42601 l=1,kcxmi
        icxmi(l)=icdelm(i,l)
42601    continue

        do 12602 l=1,kdxmi
        idxmi(l)=iddelm(i,l)
12602    continue
        imin=i
405     continue
c-проверить необходимость метки 406
406     continue

```

```

с      pause '406'
с-расчет модельных значений  $um(i)$  по  $min$  ошибке
      write(6,701) imin
      write(6,16532)
16532 format(//1x,'Модель для зависимого параметра'/)
      kprod=7
      iprod=0
      iprech=0
      do 4001 iarg=1,karg
      narg=mnarg(iarg)
      write(6,4002) narg, (anaz(narg,j1),j1=1,60)
4002  format(/1x,' аргумент-',i3/1x,60a1)
      km1=imin*m+1
      km=imin*m
      kr=imin

      in=(iarg-1)*imin+1
      ik=in+imin-1
      do 876 j=in,ik
      write(6,12451) narg,kr
12451 format(1x,'номер аргумента-',i3,' степень=',i3/1x,
*' коэффициент')
      iz1=izakm(imin,j)
      kcel=kcakm(imin,j)
      kdel=kdakm(imin,j)
      do 12561 l=1,kcel
      icr(l)=icakm(imin,j,l)
12561 continue
      do 12562 l=1,kdel
      idr(l)=idakm(imin,j,l)
12562 continue
      write(6,12563) iz1
12563 format(1x,'знак=',i2)
      write(6,12564)
12564 format(1x,'целая часть коэффициента'/)

      write(6,12565) (icr(l),l=1,kcel)
12565 format(1x,70i1)
      write(6,12566)
12566 format(1x,'дробная часть коэффициента')

      write(6,12565) (idr(l),l=1,kdel)

с      write(9,50211) akm(imin,j),narg,kr
с      write(6,50211) akm(imin,j),narg,kr

      kr=kr-1
876   continue
4001  continue
с     write(6,14007) akm(imin,km1)
с     write(9,14007) akm(imin,km1)
с     write(*,14007) akm(imin,km1)
с     pause
      iz=izakm(imin,km1)
      kcel=kcakm(imin,km1)
      kdel=kdakm(imin,km1)
      do 13401 l=1,kcel
      icr(l)=icakm(imin,km1,l)
13401 continue
      do 13402 l=1,kdel
      idr(l)=idakm(imin,km1,l)
13402 continue

```

```

        write(6,13403)
13403 format(/1x,'свободный член модели'/)

        write(6,13404) iz
13404 format(1x,'знак свободного члена=',i2)
        write(6,13405)
13405 format(1x,'целая часть свободного члена')
        write(6,12565) (icr(l),l=1,kcel)
        write(6,43405)
43405 format(1x,'дробная часть свободного члена')
        write(6,12565) (idr(l),l=1,kdel)
14007 format(5x,'+',d23.15,' ')
4007 format(5x,'+',f25.6,'')//1x,
        '*'-умножение,**'-возведение в степень'/)

c      delt=0
        izdelt=1
        kcdelt=1
        kddelt=1
        icdelt(1)=0
        iddelt(1)=0
        do 5201 i=1,m
            izwkl(i)=1
            kcwkl(i)=1
            kdwkl(i)=1
c      wkl(i)=0
            icwkl(i,1)=0
            idwkl(i,1)=0
5201    continue
        izsy=1
        kcsy=1
        kdsy=1
c      sy=0
            icsy(1)=0
            idsy(1)=0
            do 3354 i=1,n
c      ym(i)=akm(imin,km1)
            izym(i)=izakm(imin,km1)
            kcym(i)=kcakm(imin,km1)
            kdym(i)=kdakm(imin,km1)
            kcel=kcym(i)
            kdel=kdym(i)
            do 15601 l=1,kcel
                icym(i,l)=icakm(imin,km1,l)
15601    continue
            do 15602 l=1,kdel
                idym(i,l)=idakm(imin,km1,l)
15602    continue
            do 33540 j=1,m
                in=(j-1)*(imin)+1
                ik=in+imin-1
                isp=1
                do 3355 ist=in,ik
                    istep=imin-(isp-1)
                    izw=1
                    kcw=1
                    kdw=1
                    icw(1)=1
                    idw(1)=0
                    izr=izx(i,j)
                    kcr=kcx(i,j)
                    kdr=kdx(i,j)
                    do 31101 l=1,kcr

```

```

    icr(l)=icx(i,j,l)
31101 continue
    do 31102 l=1,kdr
    idr(l)=idx(i,j,l)
31102 continue
    if(istep.eq.0) goto 31103
    do 31104 isw=1,istep
    call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
    *kcr,kdr,icr,idr,
    *kcrez,kdrez,icrez,idrez,krazr)
    izw=izrez
    kcw=kcrez
    kdw=kdrez
    do 31105 l=1,kcw
    icw(l)=icrez(l)
31105 continue
    do 31106 l=1,kdw
    idw(l)=idrez(l)
31106 continue
31104 continue
31103 continue
    izr=izakm(imin,ist)
    kcr=kcakm(imin,ist)
    kdr=kdakm(imin,ist)
    do 17601 l=1,kcr
    icr(l)=icakm(imin,ist,l)
17601 continue
    do 17602 l=1,kdr
    idr(l)=idakm(imin,ist,l)
17602 continue
    call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
    *kcr,kdr,icr,idr,
    *kcrez,kdrez,icrez,idrez,krazr)
    iz1=izym(i)
    kcr=kcym(i)
    kdr=kdym(i)
    do 17301 l=1,kcr
    icr(l)=icym(i,l)
17301 continue
    do 17302 l=1,kdr
    idr(l)=idym(i,l)
17302 continue
    call slzn(iz1,izrez,iz2,kcr,kdr,icr,idr,
    *kcrez,kdrez,icrez,idrez,
    *kcl,kdl,icl,idl,krazr)
    izym(i)=iz2
    kcym(i)=kcl
    kdym(i)=kdl
    do 17303 l=1,kcl
    icym(i,l)=icl(l)
17303 continue
    do 17304 l=1,kdl
    idym(i,l)=idl(l)
17304 continue

c    ym(i)=ym(i)+akm(imin,ist)*x(i,j)**(imin-(isp-1))
    iz1=izwkl(j)
    kcel=kcwkl(j)
    kdel=kdwkl(j)
    do 18010 l=1,kcel
    icr(l)=icwkl(j,l)
18010 continue
    do 18011 l=1,kdel

```

```

      idr(1)=idwkl(j,1)
18011 continue
      izrez=izrez*izrez
      call slzn(iz1,izrez,iz2,kcel,kdel,icr,idr,
      *kcrez,kdrez,icrez,idrez,
      *kcl,kdl,icl,idl,krazr)
c      wk1(j)=wk1(j)+dabs(akm(imin,ist)*x(i,j)**(imin-(isp-1)))
      izwkl(j)=iz2
      kcwkl(j)=kcl
      kdwkl(j)=kdl
      do 18101 l=1,kcl
      icwkl(j,l)=icl(l)
18101 continue
      do 18102 l=1,kdl

      idwkl(j,1)=idl(l)
18102 continue

      isp=isp+1
3355 continue
33540 continue
      izr=izym(i)
      kcr=kcym(i)
      kdr=kdym(i)
      do 19101 l=1,kcr
      icr(l)=icym(i,l)
19101 continue
      do 19102 l=1,kdr
      idr(l)=idym(i,l)
19102 continue
      iz1=izakm(imin,kml)
      kcl=kcakm(imin,kml)
      kdl=kdakm(imin,kml)
      do 19103 l=1,kcl
      icl(l)=icakm(imin,kml,l)
19103 continue
      do 19104 l=1,kdl
      idl(l)=idakm(imin,kml,l)
19104 continue
      call wicz(izr,iz1,izrez,kcr,kdr,icr,idr,
      *kcl,kdl,icl,idl,
      *kcrez,kdrez,icrez,idrez,krazr)
c      sy=sy+ym(i)-akm(imin,kml)
      call slzn(izsy,izrez,iz2,kcsy,kdsy,icsy,idsy,
      *kcrez,kdrez,icrez,idrez,
      *kcl,kdl,icl,idl,krazr)
      izsy=iz2
      kcsy=kcl
      kdsy=kdl
      do 19201 l=1,kcl
      icsy(l)=icl(l)
19201 continue
      do 19203 l=1,kdsy
      idsy(l)=idl(l)
19203 continue
      izr=izym(i)
      kcr=kcym(i)
      kdr=kdym(i)
      do 19301 l=1,kcr
      icr(l)=icym(i,l)
19301 continue
      do 19302 l=1,kdr
      idr(l)=idym(i,l)

```

```

19302 continue
c   write(6,*) i
c   write(6,*) 'модельное y'
c   write(6,*) izr
c   write(6,*) 'целая часть'
c   write(6,87321) (icr(l),l=1,kcr)
87321 format(1x,60i1)
c   write(6,*) 'дробная часть'
c   write(6,87321) (idr(l),l=1,kdr)
      izl=izy(i)
      kc1=kcyl(i)
      kd1=kyd(i)
      do 19401 l=1,kc1
      ic1(l)=icy(i,l)
19401 continue
      do 19402 l=1,kd1
      id1(l)=idy(i,l)
19402 continue
c   write(6,*) 'исходное y'
c   write(6,*) iz1
c   write(6,*) 'целая часть'
c   write(6,87321) (ic1(l),l=1,kc1)
c   write(6,*) 'дробная часть'
c   write(6,87321) (id1(l),l=1,kd1)
      call wiczn(izr,izl,iz2,kcr,kdr,icr,idr,
      *kc1,kd1,ic1,id1,
      *kcrez,kdrez,icrez,idrez,krazr)
      izrez=iz2*iz2
c   delt=delt+dabs(ym(i)-y(i))
      call slzn(izdelt,izrez,iz2,kcdelt,kddelt,icdelt,iddelt,
      *kcrez,kdrez,icrez,idrez,
      *kc1,kd1,ic1,id1,krazr)
      izdelt=iz2
      kcdelt=kc1
      kddelt=kd1
      do 19501 l=1,kc1
      icdelt(l)=ic1(l)
19501 continue
      do 19502 l=1,kd1
      iddelt(l)=id1(l)
19502 continue

c345 format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
3354   continue
      izwtot=1
      kcwtot=1
      kdwtot=1
      icwtot(1)=0
      idwtot(1)=0
c   wkltot=0
      do 5204 i=1,m
      izr=1
      kcr=kcwkl(i)
      kdr=kdwkl(i)
      do 19451 l=1,kcr
      icr(l)=icwkl(i,l)
19451 continue
      do 19452 l=1,kdr
      idr(l)=idwkl(i,l)
19452 continue
c   wkltot=wkltot+dabs(wkl(i))
      call slzn(izwtot,izr,iz1,kcwtot,kdwtot,icwtot,idwtot,

```

```

*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
  izwtot=izl
  kcwtot=kcrez
  kdwtot=kdrez
  do 19871 l=1,kcrez
  icwtot(l)=icrez(l)
19871 continue
  do 19872 l=1,kdrez
  idwtot(l)=idrez(l)
19872 continue
5204 continue
c   write(6,5307) sy,wkltot
5307 format(/1x,'sy=',f15.5,' wkltot=',f15.5/)
c   if(wkltot.eq.0) goto 5202
  do 5203 i=1,m
  izr=izwkl(i)
  kcr=kcwkl(i)
  kdr=kdwkl(i)
  do 19671 l=1,kcr
  icr(l)=icwkl(i,l)
19671 continue
  do 19672 l=1,kdr
  idr(l)=idwkl(i,l)
19672 continue
c   wkl(i)=wkl(i)/wkltot
  call delzn(izr,izwtot,izl,kcr,kdr,icr,idr,
*kcwtot,kdwtot,icwtot,idwtot,
*kcl,kdl,icl,idl,krazr)
  izwkl(i)=izl
  kcwkl(i)=kcl
  kdwkl(i)=kdl
  do 19345 l=1,kcl
  icwkl(i,l)=icl(l)
19345 continue
  do 19346 l=1,kdl
  idwkl(i,l)=idl(l)
19346 continue
5203 continue
5202 continue
c   delt=delt/n
  call delzn(izdelt,izn,izr,kcdelt,kddelt,icdelt,iddelt,
*kcn,kdn,icn,idn,
*kcrez,kdrez,icrez,idrez,krazr)
  izdelt=izr
  kcdelt=kcrez
  kddelt=kdrez
  do 19632 l=1,kcrez
  icdelt(l)=icrez(l)
19632 continue
  do 19633 l=1,kdrez
  iddelt(l)=idrez(l)
19633 continue

702  format(1x,6f13.5)
701  format(//1x,'оптимальная степень полинома=',i3//)
c   write(6,891) delt
  write(6,*) 'средняя абсолютная ошибка'
  write(6,*) 'целая часть ошибки'
  write(6,19711) (icdelt(l),l=1,kcdelt)
19711 format(1x,70i1)
  write(6,*) 'дробная часть ошибки'
  write(6,19711) (iddelt(l),l=1,kddelt)

```

```

с write(6,5401)
5401 format(/18x,'Таблица-Вклады параметров-аргументов'/)
с write(6,5402)
5402 format(11x,51('-'))
с write(6,5403)
5403 format(11x,':Номер:',6x,'Название параметра',6x,':',3x,
*'Вклад',4x,':')
с write(6,5402)
prob=' '
do 5404 i=1,m
narg=mnarg(i)
write(6,5405) narg,(anaz(narg,j),j=1,60)
5405 format(1x,'номер аргумента=',i3/1x,60a1)
write(6,*) 'целая часть вклада'
kcel=kcwkl(i)
write(6,19711) (icwkl(i,l),l=1,kcel)
write(6,*) 'дробная часть вклада'
kdel=kdwkl(i)
write(6,19711) (idwkl(i,l),l=1,kdel)

5404 continue
891 format(/1x,'средняя абсолютная ошибка=',f36.18/)
9988 continue
2001 continue
close(4,status='keep')
close(6,status='keep')
close(5,status='keep')
close(7,status='keep')
end
$large
subroutine mnk(n,m,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
*izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,igau,krazr)
с-метод наименьших квадратов
с dimension x(100,20),y(100),ak(100),ym(100),c(100,100),f(100)
с dimension weko(100)
с double precision x,y,ak,ym,c,f,weko,sum
integer*2 izf(31),kcf(31),kdf(31),icf(31,20),idf(31,20)
integer*2 icw(400),idw(400),izx(50,10),kcx(50,10),kdx(50,10)
integer*2 icx(50,10,20),idx(50,10,20),icrez(400),idrez(400),
*izy(50),kcy(50),kdy(50),icy(50,20),idy(50,20),icr(400),idr(400),
*izc(31,31),kcc(31,31),kdc(31,31),icc(31,31,20),idc(31,31,20),
*icw1(400),idw1(400),ic1(400),id1(400),icn(400),idn(400)
integer*2 izak(31),kcak(31),kdak(31),icak(31,20),idak(31,20)
integer s,p
с-с(50,50),матрица системы нормальных уравнений
с-f(50),вектор свободных членов
с-ak(50),коэффициенты полиномов по аргументам
с-x(50,10),матрица аргументов
с-y(50),массив значений функции
с-ym(50),массив модельных значений функции
с write(6,*) ' Метод наименьших квадратов'
с do 1 i=1,n

с write(6,123) i,x(i),i,y(i)
123 format(1x,'x(',i3,')=',f13.5,' y(',i3,')=',f13.5)
1 continue
с-определение матрицы с
с-i-номер наблюдения,строчки
с-j-номер параметра-аргумента
с-l-номер коэффициента,степени
с-s-номер степени,p-номер параметра
с-m-количество параметров-аргументов

```

```

с-k, порядок полиномов
с-n, число значений функции
с      write(6,*) 'передача в mnk'
с      write(6,*) 'матрица x'
с      do 34521 i=1,n
с      do 34522 j=1,m
с      write(6,*) i,j
с      write(6,*) 'целая часть x'
с      izn=izx(i,j)
с      write(6,*) izn
с      kcel=kcx(i,j)
с      kdel=kdx(i,j)
с      write(6,34523) (icx(i,j,l),l=1,kcel)
34523 format(1x,50i1)
с      write(6,*) 'дробная часть x'
с      write(6,34523) (idx(i,j,l),l=1,kdel)
34522 continue
34521 continue
с      write(6,*) 'вектор y'
с      do 34545 i=1,n
с      kcel=kcy(i)
с      kdel=kdy(i)
с      write(6,*) izy(i)
с      write(6,*) 'целая часть y'
с      write(6,34523) (icy(i,l),l=1,kcel)
с      write(6,*) 'дробная часть y'
с      write(6,34523) (idy(i,l),l=1,kdel)
34545 continue
с      stop
с      k1=k+1
с      pause 'mnk1'
с      do 2 p=1,m
с      do 282 s=1,k
с      izf((p-1)*k+s)=1
с      kcf((p-1)*k+s)=1
с      kdf((p-1)*k+s)=1
с      icf((p-1)*k+s,1)=0
с      idf((p-1)*k+s,1)=0
с      f((p-1)*k+s)=0
с      do 2002 i=1,n
с      izw=1
с      kcw=1
с      kdw=1
с      icw(1)=1
с      idw(1)=0
с      izr=izx(i,p)
с      kcr=kcx(i,p)
с      kdr=kdx(i,p)
с      do 12101 l=1,kcr
с      icr(1)=icx(i,p,l)
12101 continue
с      do 12102 l=1,kdr
с      idr(1)=idx(i,p,l)
12102 continue
с      istep=k-(s-1)
с      if(istep.eq.0) goto 32103
с      do 12104 l=1,istep
с      pause '1'
с      write(6,*) 'p1'
с      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
с      write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez

```

```

c      pause '2'
      izw=izrez
      kcw=kcrez
      kdw=kdrez
      do 12201 lp=1,kcrez
12201  icw(lp)=icrez(lp)
      continue
      do 12202 lp=1,kdrez
12202  idw(lp)=idrez(lp)
      continue
12104  continue
32103  continue
c      pause '32103'
      izr=izy(i)
      kcr=kcy(i)
      kdr=kdy(i)
      do 13291 l=1,kcr
13291  icr(l)=icy(i,l)
      continue
      do 13282 l=1,kdr
13282  idr(l)=idy(i,l)
      continue
c      pause '3'
c      write(6,*) 'p3'
      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
c      write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez
c      pause '4'
      ilf=(p-1)*k+s
      izr=izf(ilf)
      kcr=kcf(ilf)
      kdr=kdf(ilf)
      do 14301 l=1,kcr
14301  icr(l)=icf(ilf,l)
      continue
      write(*,*) icr(l)
      do 14302 l=1,kdr
14302  idr(l)=idf(ilf,l)
      continue
      write(*,*) idr(l)
c      pause '5 na'
c      write(6,*) 'первое слагаемое'
c      write(6,*) izr,kcr,kdr
c      write(6,*) izr,kcr,kdr
c      pause '5_1'
c      write(6,*) izr
c      write(6,49701) izr
49701  format(1x,'знак=',i3/)
c      write(6,*) 'целая часть'
c      write(6,*) (icr(l),l=1,kcr)
c      pause '5_1_1'
c      write(6,*) (icr(l),l=1,kcr)
c      pause
49702  format(1x,50i1)
c      write(6,*) 'дробная часть'
c      write(6,*) (idr(l),l=1,kdr)
49703  format(1x,50i1)
c      write(6,*) 'второе слагаемое'
c      write(6,49701) izrez
c      write(6,*) izrez,kcrez,kdrez
c      pause '5_2'
c      write(6,*) 'целая часть'

```

```

c      write(6,*) (icrez(l),l=1,kcrez)
c      write(6,*) 'дробная часть'
c      write(6,*) (idrez(l),l=1,kdrez)
c      pause '5'
      call slzn(izr,izrez,izl,kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,
      *kcl,kdl,icl,idl,krazr)
c      pause '6'
c      f((p-1)*k+s) = f((p-1)*k+s)+y(i)*x(i,p)**(k-(s-1))
      izf(ilf)=izl
      kcf(ilf)=kcl
      kdf(ilf)=kdl
      do 15401 l=1,kcl
15401  icf(ilf,l)=icl(l)
      do 15402 l=1,kdl
15402  idf(ilf,l)=idl(l)
2002  continue
c      pause '2002'
      do 212 j=1,m
      do 211 l=1,k
      ilc=(p-1)*k+s
      jlcl=(j-1)*k+l
c      c((p-1)*k+s,(j-1)*k+l)=0
      izc(ilc,jlcl)=1
      kcc(ilc,jlcl)=1
      kdc(ilc,jlcl)=1
      icc(ilc,jlcl,1)=0
      idc(ilc,jlcl,1)=0
      do 3 i=1,n
c      c((p-1)*k+s,(j-1)*k+l) = c((p-1)*k+s,(j-1)*k+l)+
c      *x(i,j)**(k-(l-1))*x(i,p)**(k-(s-1))
      izw=1
      kcw=1
      kdw=1
      icw(1)=1
      idw(1)=0
      istep=k-(l-1)
      izr=izx(i,j)
      kcr=kcx(i,j)
      kdr=kdx(i,j)
      do 14201 lp=1,kcr
14201  icr(lp)=icx(i,j,lp)
      do 14202 lp=1,kdr
14202  idr(lp)=idx(i,j,lp)
14202  continue
      if(istep.eq.0) goto 16101
      do 16102 lpr=1,istep
c      pause '7'
      call ymnzn(izw,izr,izl,kcw,kdw,icw,idw,
      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '8'
      izw=izl
      kcw=kcrez
      kdw=kdrez
      do 17101 lp=1,kcw
17101  icw(lp)=icrez(lp)
      do 17102 lp=1,kdw
17102  idw(lp)=idrez(lp)

```

```

17102 continue
16102 continue
16101 continue
      istep=k-(s-1)
      izw1=1
      kcw1=1
      kdwl=1
      icw1(1)=1
      idw1(1)=0
      izr=izx(i,p)
      kcr=kcx(i,p)
      kdr=kdx(i,p)
      do 18101 lp=1,kcr
18101 continue
      icr(lp)=icx(i,p,lp)
      do 18102 lp=1,kdr
18102 continue
      idr(lp)=idx(i,p,lp)
      if(istep.eq.0) goto 18301
      do 18302 lpr=1,istep
c      pause '9'
      call ymnzn(izw1,izr,izrez,kcw1,kdwl,icw1,idw1,
      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '10'
      izw1=izrez
      kcw1=kcrez
      kdwl=kdrez
      do 13201 lp=1,kcrez
13201 continue
      icw1(lp)=icrez(lp)
      do 63201 lp=1,kdrez
63201 continue
      idw1(lp)=idrez(lp)
18302 continue
18301 continue
c      pause '11'
      call ymnzn(izw,izw1,izrez,kcw,kdw,icw,idw,
      *kcw1,kdwl,icw1,idw1,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '12'
      izr=izc(ilc,jlc)
      kcr=kcc(ilc,jlc)
      kdr=kdc(ilc,jlc)
      do 17701 lp=1,kcr
17701 continue
      icr(lp)=icc(ilc,jlc,lp)
      do 17702 lp=1,kdr
17702 continue
      idr(lp)=idc(ilc,jlc,lp)
c      pause '13'
      call slzn(izrez,izr,izl,kcrez,kdrez,icrez,idrez,
      *kcr,kdr,icr,idr,
      *kcl,kdl,icl,idl,krazr)
c      pause '14'
      izc(ilc,jlc)=izl
      kcc(ilc,jlc)=kcl
      kdc(ilc,jlc)=kdl
      do 16701 lp=1,kcl

      icc(ilc,jlc,lp)=icl(lp)
16701 continue
      do 16702 lp=1,kdl

```

```

        idc(ilc,jlc,lp)=idl(lp)
16702 continue
3      continue
      istr=(p-1)*k+s
      isto=(j-1)*k+1
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
3311  format(1x,'p=',i3,' s=',i3,' j=',i3,' l=',i3/1x,
*' istr=',i3,' isto=',i3,' c=',f15.5,' f=',f15.5/)
211   continue
212   continue
      ilc=(p-1)*k+s
      jlc=(m-1)*k+k+1
      izc(ilc,jlc)=1
      kcc(ilc,jlc)=1
      kdc(ilc,jlc)=1
      icc(ilc,jlc,1)=0
      idc(ilc,jlc,1)=0
c      c((p-1)*k+s,(m-1)*k+k+1)=0
      do 703 i=1,n
      istep=k-(s-1)
      izw=1
      kcw=1
      kdw=1
      icw(1)=1
      idw(1)=0
      izr=izx(i,p)
      kcr=kcx(i,p)
      kdr=kdx(i,p)
      do 18401 lp=1,kcr
      icr(lp)=icx(i,p,lp)
18401 continue
      do 18402 lp=1,kdr
      idr(lp)=idx(i,p,lp)
18402 continue
      if(istep.eq.0) goto 18403

      do 18404 lpr=1,istep
c      pause '15'
      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
c      pause '16'
      izw=izrez
      kcw=kcrez
      kdw=kdrez
      do 16785 lp=1,kcrez
      icw(lp)=icrez(lp)
16785 continue
      do 16786 lp=1,kdrez
      idw(lp)=idrez(lp)
16786 continue
18404 continue
18403 continue
      izr=izc(ilc,jlc)
      kcr=kcc(ilc,jlc)
      kdr=kdc(ilc,jlc)
      do 11301 lp=1,kcr
      icr(lp)=icc(ilc,jlc,lp)
11301 continue
      do 11302 lp=1,kdr
      idr(lp)=idc(ilc,jlc,lp)
11302 continue
c      pause '17'

```

```

    call slzn(izw,izr,izrez,kcw,kdw,icw,idw,
*kcr,kdr,icr,idr,
*kcrez,kdrez,icrez,idrez,krazr)
c    pause '18'
    izc(ilc,jlc)=izrez
    kcc(ilc,jlc)=kcrez
    kdc(ilc,jlc)=kdrez
    do 13781 lp=1,kcrez
    icc(ilc,jlc,lp)=icrez(lp)

13781 continue
    do 13782 lp=1,kdrez
    idc(ilc,jlc,lp)=idrez(lp)
13782 continue

c    c((p-1)*k+s,(m-1)*k+k+1)=c((p-1)*k+s,(m-1)*k+k+1)+
c    *x(i,p)**(k-(s-1))

703  continue
    istr=(p-1)*k+s
    isto=(m-1)*k+k+1
c    write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
282  continue
2    continue
    inf=(m-1)*k+k+1
    izf(inf)=1
    kcf(inf)=1
    kdf(inf)=1
    icf(inf,1)=0
    idf(inf,1)=0
c    f((m-1)*k+k+1)=0
    do 12002 i=1,n
    izr=izf(inf)
    kcr=kcf(inf)
    kdr=kdf(inf)
    do 14781 lp=1,kcr
    icr(lp)=icf(inf,lp)
14781 continue
    do 14782 lp=1,kdr
    idr(lp)=idf(inf,lp)
14782 continue
    iz1=izy(i)
    kcl=kcy(i)
    kdl=kdy(i)
    do 14785 lp=1,kcl
    icl(lp)=icy(i,lp)
14785 continue
    do 14786 lp=1,kdl
    idl(lp)=idy(i,lp)
14786 continue
c    pause '19'
    call slzn(izr,iz1,izrez,kcr,kdr,icr,idr,
*kcl,kdl,icl,idl,
*kcrez,kdrez,icrez,idrez,krazr)
c    pause '20'
    izf(inf)=izrez
    kcf(inf)=kcrez
    kdf(inf)=kdrez
    do 44871 lp=1,kcrez
    icf(inf,lp)=icrez(lp)
44871 continue
    do 14872 lp=1,kdrez
    idf(inf,lp)=idrez(lp)

```

```

14872 continue
c      f((m-1)*k+k+1)= f((m-1)*k+k+1)+y(i)
12002 continue
      do 4212 j=1,m
      do 4211 l=1,k
      iccl=(m-1)*k+k+1
      jcl=(j-1)*k+1
      izc(iccl,jcl)=1
      kcc(iccl,jcl)=1
      kdc(iccl,jcl)=1
      icc(iccl,jcl,1)=0
      idc(iccl,jcl,1)=0
c      c((m-1)*k+k+1,(j-1)*k+1)=0
      do 371 i=1,n
      istep=k-(l-1)
      izw=1
      kcw=1
      kdw=1
      icw(1)=1
      idw(1)=0
      izr=izx(i,j)
      kcr=kcx(i,j)
      kdr=kdx(i,j)
      do 23671 lp=1,kcr
      icr(lp)=icx(i,j,lp)
23671 continue
      do 23672 lp=1,kdr
      idr(lp)=idx(i,j,lp)
23672 continue
      if(istep.eq.0) goto 23673
      do 23674 lpr=1,istep
c      pause '21'
      call ymnzn(izw,izr,izrez,kcw,kdw,icw,idw,
      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '22'
      izw=izrez
      kcw=kcrez
      kdw=kdrez
      do 24301 lp=1,kcrez
      icw(lp)=icrez(lp)
24301 continue
      do 24302 lp=1,kdrez
      idw(lp)=idrez(lp)
24302 continue
23674 continue
23673 continue
      izr=izc(iccl,jcl)
      kcr=kcc(iccl,jcl)
      kdr=kdc(iccl,jcl)
      do 25401 lp=1,kcr
      icr(lp)=icc(iccl,jcl,lp)
25401 continue
      do 25402 lp=1,kdr
      idr(lp)=idc(iccl,jcl,lp)
25402 continue
c      pause '23'
      call slzn(izw,izr,izrez,kcw,kdw,icw,idw,
      *kcr,kdr,icr,idr,
      *kcrez,kdrez,icrez,idrez,krazr)
c      pause '24'
      izc(iccl,jcl)=izrez
      kcc(iccl,jcl)=kcrez

```

```

        kdc(iccl,jcl)=kdrez
        do 26701 lp=1,kcrez
            icc(iccl,jcl,lp)=icrez(lp)
26701 continue
        do 26702 lp=1,kdrez
            idc(iccl,jcl,lp)=idrez(lp)
26702 continue
c      c((m-1)*k+k+1,(j-1)*k+1)=c((m-1)*k+k+1,(j-1)*k+1)+
c      *x(i,j)**(k-(l-1))
371   continue
        istr=(m-1)*k+k+1
        isto=(j-1)*k+1
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
c3311   format(1x,'p=',i3,' s=',i3,' j=',i3,' l=',i3/1x,
c      *'istr=',i3,' isto=',i3,' c=',f15.5,' f=',f15.5/)
4211   continue
4212   continue
        iccl=(m-1)*k+k+1
        izc(iccl,iccl)=izn
        kcc(iccl,iccl)=kcn
        kdc(iccl,iccl)=kdn
        do 27101 lp=1,kcn
            icc(iccl,iccl,lp)=icn(lp)
27101 continue
        do 27102 lp=1,kdn
            idc(iccl,iccl,lp)=idn(lp)
27102 continue
c      c((m-1)*k+k+1,(m-1)*k+k+1)=n

        istr=(m-1)*k+k+1
        isto=(m-1)*k+k+1
c      write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)

        kraz=m*k+1
c      write(5,*) ' матрица c'
        open(9,file='promprow',status='Unknown')
c      write(5,*) kraz
        write(9,*) kraz,krazr
        do 56 i=1,kraz
            do 1234 j=1,kraz
c      write(5,4581) i,j
4581   format(1x,'ic=',i3,' jc=',i3)
c      write(5,*) 'целая часть'
            izr=izc(i,j)
            kcel=kcc(i,j)
            kdel=kdc(i,j)

            write(9,33357) izr,kcel,kdel,(icc(i,j,jp),jp=1,kcel),
            *(idc(i,j,jp),jp=1,kdel)
33357 format(3i2,80i1)
1234   continue
56     continue
57     format(1x,70i1)
c      write(6,*) ' вектор свободных членов'
c4981 continue
        do 56701 i=1,kraz
            izr=izf(i)
            kcel=kcf(i)
            kdel=kdf(i)
c      write(6,56702) i
56702 format(1x,'fi=',i6)
c      write(6,*) 'целая часть'

```

```
        write(9,33357) izr,kcel,kdel,(icf(i,ip),ip=1,kcel),
        *(idf(i,ip),ip=1,kdel)
56701 continue
        close(9,status='keep')
        kg=k*m+1
c       call gay(kg,c,f,ak,igau)
c       pause 'pered gaydl'
        call gaydl(kg,izc,kcc,kdc,icc,idc,izf,kcf,kdf,icf,idf,
        *izak,kcak,kdak,icak,idak,igau,krazr)
c       pause 'posle gaydl'
        kg=k*m+1
        call prwg(kg,izc,kcc,kdc,icc,idc,
        *izf,kcf,kdf,icf,idf,izak,kcak,kdak,icak,idak,igau,krazr)

        if(igau.eq.0) return
        return
        end
```

## Набор gaydl.for

\$large

c-метод Гаусса решения систем с выбором главного элемента, длинная арифметика

c-ac(i,j,l)-целая часть элемента i,j матрицы системы

c-ad(I,j,l)-дробная часть элемента I,j матрицы системы

c-nc(I,j)-количество разрядов целой части элемента I,j

c-nd(I,j)-количество разрядов дробной части элемента I,j

c-znak(I,j)-знак элемента I,j

c-bc(I,l)-целая часть правой части i-ого уравнения

c-bd(I,l)-дробная часть правой части i-го уравнения

c-nbc(i)-длина целой части правой части i-го уравнения

c-nbd(i)-длина дробной части правой части i-го уравнения

c-znp(i)-знак правой части i-ого уравнения

c-xc(I,l)-целая часть i-го неизвестного

c-xd(I,l)-дробная часть i-го неизвестного

c-nxc(i)-длина целой части i-ого неизвестного

c-nxd(i)-длина дробной части i-ого неизвестного

c-znx(i)-знак i-ой неизвестной

```
      subroutine gaydl(n,znak,nc,nd,ac,ad,znp,nbc,nbd,bc,bd,znx,nxc,nxd,
```

```
      *xc,xd,igau,krazr)
```

```
      dimension mnper(100)
```

```
      integer*2 znak(31,31),nc(31,31),nd(31,31),tc(400),td(400),
```

```
      *ac(31,31,20),ad(31,31,20),znp(31),nbc(31),nbd(31)
```

```
      integer*2 bc(31,20),bd(31,20),znx(31),nxc(31),nxd(31)
```

```
      integer*2 xc(31,20),xd(31,20)
```

```
      integer*2 anc(400),andr(400),rezc(400),rezd(400),
```

```
      *pc(400),pd(400),rc1(400),rd1(400)
```

```
      integer*2 rc(400),rd(400)
```

```
  c      integer znak,ac,ad,znp,bc,bd,znx,xc,xd,rezc,rezd,rc,rd
```

```
      integer zn
```

```
      do 38247 i=1,n
```

```
38247 mnper(i)=i
```

```
      if(krazr.le.50) goto 8855
```

```
      write(6,*) 'количество разрядов>50,нельзя!'
```

```
      stop
```

```
8855 continue
```

```
      nis=n
```

```
      i=1
```

```
81      j=i
```

```
  c      pause 'pered nimax'
```

```
      call nimax(i,j,n,nc,nd,znak,ac,ad,nbc,nbd,znp,bc,bd,
```

```
      *mnper,krazr)
```

```
  c      pause 'posle nimax'
```

```
  c      t=a(i,i)
```

```
      nct=nc(i,i)
```

```
      ndt=nd(i,i)
```

```
      iznt=znak(i,i)
```

```
      do 41026 ld=1,nct
```

```
41026 tc(ld)=ac(i,i,ld)
```

```
      do 41027 ld=1,ndt
```

```
41027 td(ld)=ad(i,i,ld)
```

```
      do 41028 ld=1,100
```

```
      anc(ld)=0
```

```
      andr(ld)=0
```

```
41028 continue
```

```
      nolc=100
```

```
      nold=100
```

```
      iznol=1
```

```
      do 41029 ld=1,nct
```

```
      if(tc(ld).ne.anc(ld)) goto 2
```

```
41029 continue
```

```
      do 41031 ld=1,ndt
```

```

        if(td(ld).ne.andr(ld)) goto 2
41031 continue
c      goto 2
c      if(dabs(t).gt.0.1d-38) goto 2
        l=i+1
c4     if(dabs(a(l,j)).gt.0.1d-38) goto 3
4      continue
        ncl=nc(l,j)
        ndl=nd(l,j)
        iznl=znak(l,j)
        do 41032 ld=1,ncl
        if(ac(l,j,ld).ne.anc(ld)) goto 3
41032 continue
        do 41033 ld=1,ndl
        if(ad(l,j,ld).ne.andr(ld)) goto 3
41033 continue
        l=l+1
        if(l.le.n) goto 4
        write(6,*) 'матрица вырожденная'
c      write(5,*) 'столбик диагонального элемента'
        do 56721 l=i,n
c      write(5,56722) l,a(l,i)
56722 format(1x,'i=',i3,' a=',d36.18)
56721 continue
        igau=0
        return
3      continue
        do 5 js=1,n
c      tz=a(l,js)
c      a(l,js)=a(i,js)
c      a(i,js)=tz
        nct=nc(l,js)
        ndt=nd(l,js)
        izt=znak(l,js)
        do 41034 ld=1,nct
41034 tc(ld)=ac(l,js,ld)
        do 41035 ld=1,ndt
41035 td(ld)=ad(l,js,ld)
        nc(l,js)=nc(i,js)
        nd(l,js)=nd(i,js)
        znak(l,js)=znak(i,js)
        nlc=nc(i,js)
        nld=nd(i,js)
        do 41036 ld=1,nlc
41036 ac(l,js,ld)=ac(i,js,ld)
        do 41037 ld=1,nld
41037 ad(l,js,ld)=ad(i,js,ld)
        nc(i,js)=nct
        nd(i,js)=ndt
        znak(i,js)=izt
        do 41038 ld=1,nct
41038 ac(i,js,ld)=tc(ld)
        do 41039 ld=1,ndt
41039 ad(i,js,ld)=td(ld)

5      continue
        nct=nc(i)
        ndt=nd(i)
        izt=znп(i)
        do 41040 ld=1,nct
41040 tc(ld)=bc(i,ld)
        do 41041 ld=1,ndt
41041 td(ld)=bd(i,ld)

```

```

    nbc(i)=nbc(l)
    nbd(i)=nbd(l)
    znp(i)=znp(l)
    n1c=nbc(l)
    n1d=nbd(l)
    do 41042 ld=1,n1c
41042 bc(i,ld)=bc(l,ld)
    do 41043 ld=1,n1d
41043 bd(i,ld)=bd(l,ld)
    nbc(l)=nct
    nbd(l)=ndt
    znp(l)=izt
    do 41044 ld=1,nct
41044 bc(l,ld)=tc(ld)
    do 41045 ld=1,ndt
    bd(l,ld)=td(ld)
41045 continue
    nct=nc(i,i)
    ndt=nd(i,i)
    iznt=znak(i,i)
    do 41046 ld=1,nct
41046 tc(ld)=ac(i,i,ld)
    do 41047 ld=1,ndt
41047 td(ld)=ad(i,i,ld)
c    t=a(i,i)
2    continue
c    write(5,15671) i
15671 format(1x,'i do=',i5)
c    write(5,4457) (a(i,js),js=1,n)
2222 continue
c    a(i,j)=a(i,j)/t
    ncr=nc(i,j)
    ndr=nd(i,j)
    izr=znak(i,j)
    do 41048 ld=1,ncr
41048 rc(ld)=ac(i,j,ld)
    do 41049 ld=1,ndr
41049 rd(ld)=ad(i,j,ld)
c    write(5,*) '41049 до delzn'
c    write(6,*) 'прямой ход, деление левой части a(i,j)/t'
c    write(6,*) 'делимое a(i,j) '
c    write(6,1891) i,j
1891 format(1x,'i=',i3,' j=',i3)
c    write(6,1371) izr
c    write(6,1892) ncr,ndr
1892 format(1x,'длина целой части=',i3,' длина дробной части=',i3)
c    write(6,1372) (rc(ld),ld=1,ncr)
c    write(6,1373) (rd(ld),ld=1,ndr)
c    write(6,*) 'делитель t'
c    write(6,1371) iznt
c    write(6,1892) nct,ndt
c    write(6,1372) (tc(ld),ld=1,nct)
c    write(6,1373) (td(ld),ld=1,ndt)
    call delzn(izr,iznt,izrez,ncr,ndr,rc,rd,nct,ndt,tc,td,
    *kcrez,kdrez,rezc,rezd,krazr)
c    write(6,*) 'частное a(i,j)=a(i,j)/t'
c    write(6,1371) izrez
c    write(6,1892) kcrez,kdrez
c    write(6,1372) (rezc(ld),ld=1,kcrez)
c    write(6,1373) (rezd(ld),ld=1,kdrez)
c    write(5,*) '41049 после delzn'
    nc(i,j)=kcrez
    nd(i,j)=kdrez

```

```

        znak(i,j)=izrez
        do 41050 ld=1,kcrez
41050 ac(i,j,ld)=rezc(ld)
        do 41051 ld=1,kdrez
41051 ad(i,j,ld)=rezd(ld)
        j=j+1
        if(j.le.n) goto 2222
c      write(5,15672) i
15672 format(1x,'i после',i5)
c      write(5,4457) (a(i,js),js=1,n)
c      write(5,15673) b(i)
15673 format(1x,'b(i) до=',d30.16)
c      b(i)=b(i)/t
        ncr=ncb(i)
        ndr=ndb(i)
        izr=znp(i)
        do 41052 ld=1,ncr
41052 rc(ld)=bc(i,ld)
        do 41053 ld=1,ndr
41053 rd(ld)=bd(i,ld)
c      write(5,*) '41053 до delzn'
c      write(6,*) 'прямой ход, деление правой части'
c      write(6,1897) i
1897 format(1x,'номер правой части=',i3)
c      write(6,*) 'делимое b(i) '
c      write(6,1371) izr
c      write(6,1892) ncr,ndr
c      write(6,1372) (rc(ld),ld=1,ncr)
c      write(6,1373) (rd(ld),ld=1,ndr)
c      write(6,*) 'делитель t'
c      write(6,1371) iznt
c      write(6,1892) nct,ndt
c      write(6,1372) (tc(ld),ld=1,nct)
c      write(6,1373) (td(ld),ld=1,ndt)
        call delzn(izr,iznt,izrez,ncr,ndr,rc,rd,nct,ndt,tc,td,
        *kcrez,kdrez,rezc,rezd,krazr)
c      write(6,*) 'частное b(i)=b(i)/t'
c      write(6,1371) izrez
c      write(6,1892) kcrez,kdrez
c      write(6,1372) (rezc(ld),ld=1,kcrez)
c      write(6,1373) (rezd(ld),ld=1,kdrez)
c      write(5,*) '41053 после delzn'
        nbc(i)=kcrez
        nbd(i)=kdrez
        znp(i)=izrez
        do 41054 ld=1,kcrez
41054 bc(i,ld)=rezc(ld)
        do 41055 ld=1,kdrez
41055 bd(i,ld)=rezd(ld)
c      write(5,15674) b(i)
15674 format(1x,'b(i) после=',d30.16)
        k=i+1
71      continue
c      write(5,15681) k
15681 format(1x,'k до=',i5)
c      write(5,4457) (a(k,js),js=1,n)

7111  continue
        ncp=nc(k,i)
        ndp=nd(k,i)
        izp=znak(k,i)
        do 41056 ld=1,ncp

```

```

41056 pc(ld)=ac(k,i,ld)
      do 41057 ld=1,ndp
41057 pd(ld)=ad(k,i,ld)
c     p=a(k,i)
      j=i
7     continue
c a(k,j)=a(k,j)-p*a(i,j)
      ncr=nc(i,j)
      ndr=nd(i,j)
      izr=znak(i,j)
      do 41058 ld=1,ncr
41058 rc(ld)=ac(i,j,ld)
      do 41059 ld=1,ndr
41059 rd(ld)=ad(i,j,ld)
c     write(6,*) 'прямой ход,умножение левой части'
c     write(6,2017) k,i,j
2017 format(1x,'уравнение k=',i3,' i=',i3,' столбик j=',i3)
c     write(6,*) 'множитель p=a(k,i)'
c     write(6,1371) izp
c     write(6,1892) ncp,ndp
c     write(6,1372) (pc(ld),ld=1,ncp)
c     write(6,1373) (pd(ld),ld=1,ndp)
c     write(6,*) 'множитель a(i,j)'
c     write(6,1371) izr
c     write(6,1892) ncr,ndr
c     write(6,1372) (rc(ld),ld=1,ncr)
c     write(6,1373) (rd(ld),ld=1,ndr)
      call ymnzn(izp,izr,izrez,ncp,ndp,pc,pd,ncr,ndr,rc,rd,
*kcrcz,kdrcz,rezc,rezd,krazr)
c     write(6,*) 'произведение p*a(i,j)'
c     write(6,1371) izrez
c     write(6,1892) kcrez,kdrez
c     write(6,1372) (rezc(ld),ld=1,kcrez)
c     write(6,1373) (rezd(ld),ld=1,kdrez)
      ncr=kcrez
      ndr=kdrez
      izr=izrez
      do 41060 ld=1,ncr
41060 rc(ld)=rezc(ld)
      do 41061 ld=1,ndr
41061 rd(ld)=rezd(ld)
      ncr1=nc(k,j)
      ndr1=nd(k,j)
      izr1=znak(k,j)
      do 41062 ld=1,ncr1
41062 rcl(ld)=ac(k,j,ld)
      do 41063 ld=1,ndr1
41063 rd1(ld)=ad(k,j,ld)
c     write(5,*) '41063 do wiczn'
c     write(6,*) 'прямой ход,до вычитания,левая часть'
c     write(6,2001) k,i,j
2001 format(/1x,'k=',i3,' i=',i3,' j=',i3)
c     write(6,*) 'уменьшаемое a(k,j)'
c     write(6,1371) izr1
c     write(6,1892) ncr1,ndr1
c     write(6,1372) (rcl(l),l=1,ncr1)
c     write(6,1373) (rd1(l),l=1,ndr1)
c     write(6,*) 'вычитаемое p*a(i,j)'
c     write(6,1371) izr
c     write(6,1892) ncr,ndr
c     write(6,1372) (rc(l),l=1,ncr)
c     write(6,1373) (rd(l),l=1,ndr)
      call wiczn(izr1,izr,izrez,ncr1,ndr1,rcl,rd1,ncr,ndr,rc,

```

```

*rd,kcrez,kdrez,rezc,rezd,krazr)
c   write(6,*) 'прямой ход,после вычитания,левая часть'
c   write(6,2001) k,i,j
c   write(6,*) 'разность a(k,j)=a(k,j)-p*a(i,j) '
c   write(6,1371) izrez
c   write(6,1892) kcrez,kdrez
c   write(6,1372) (rezc(1),l=1,kcrez)
c   write(6,1373) (rezd(1),l=1,kdrez)
c   write(5,*) '41063 после wiczn'
      nc(k,j)=kcrez
      nd(k,j)=kdrez
      znak(k,j)=izrez
      do 41064 ld=1,kcrez
41064 ac(k,j,ld)=rezc(ld)
      do 41065 ld=1,kdrez
41065 ad(k,j,ld)=rezd(ld)
      j=j+1
      if(j.le.n) goto 7
c   write(5,15682) k
15682 format(1x,'k после=',i5)
c   write(5,4457) (a(k,js),js=1,n)
4457 format(1x,2d30.16)
c   write(5,15683) b(k)
15683 format(1x,'b(k) до=',d30.16)

c   b(k)=b(k)-p*b(i)
      ncr=ncb(i)
      ndr=ncd(i)
      izr=znp(i)
      do 41066 ld=1,ncr
41066 rc(ld)=bc(i,ld)
      do 41067 ld=1,ndr
41067 rd(ld)=bd(i,ld)
c   write(6,*) 'прямой ход,умножение правой части'
c   write(6,2051) i
2051 format(1x,'уравнение=',i3)
c   write(6,*) 'множитель p'
c   write(6,1371) izp
c   write(6,1892) ncp,ndp
c   write(6,1372) (pc(ld),ld=1,ncp)
c   write(6,1373) (pd(ld),ld=1,ndp)
c   write(6,*) 'множитель b(i) '
c   write(6,1371) izr
c   write(6,1892) ncr,ndr
c   write(6,1372) (rc(ld),ld=1,ncr)
c   write(6,1373) (rd(ld),ld=1,ndr)
      call ymnzn(izp,izr,izrez,ncp,ndp,pc,pd,ncr,ndr,rc,rd,
*kcrez,kdrez,rezc,rezd,krazr)
c   write(6,*) 'произведение,прямой ход,правая часть p*b(i) '
c   write(6,1371) izrez
c   write(6,1892) kcrez,kdrez
c   write(6,1372) (rezc(ld),ld=1,kcrez)
c   write(6,1373) (rezd(ld),ld=1,kdrez)
      ncr=kcrez
      ndr=kdrez
      izr=izrez
      do 41068 ld=1,kcrez
41068 rc(ld)=rezc(ld)
      do 41069 ld=1,kdrez
41069 rd(ld)=rezd(ld)
      ncr1=ncb(k)
      ndr1=ncd(k)
      izr1=znp(k)

```

```

do 41070 ld=1,ncr1
41070 rc1(ld)=bc(k,ld)
do 41071 ld=1,ndr1
41071 rd1(ld)=bd(k,ld)
c write(5,*) '41071 do wiczn'
c write(6,*) 'прямой ход,вычитание правой части'
c write(6,2341) k
2341 format(1x,'правая часть,прямой ход,вычитание k=',i3)
c write(6,*) 'уменьшаемое b(k) '
c write(6,1371) izr1
c write(6,1892) ncr1,ndr1
c write(6,1372) (rc1(ld),ld=1,ncr1)
c write(6,1373) (rd1(ld),ld=1,ndr1)
c write(6,*) 'вычитаемое p*b(i) '
c write(6,1371) izr
c write(6,1892) ncr,ndr
c write(6,1372) (rc(ld),ld=1,ncr)
c write(6,1373) (rd(ld),ld=1,ndr)
call wiczn(izr1,izr,izrez,ncr1,ndr1,rc1,rd1,ncr,ndr,rc,rd,
*kcrez,kdrez,rezc,rezd,krazr)
c write(6,*) 'прямой ход,разность b(k)=b(k)-p*b(i) '
c write(6,1371) izrez
c write(6,1892) kcrez,kdrez
c write(6,1372) (rezc(ld),ld=1,kcrez)
c write(6,1373) (rezd(ld),ld=1,kdrez)
c write(5,*) '41071 после wiczn'
nbc(k)=kcrez
nbd(k)=kdrez
znp(k)=izrez
do 41072 ld=1,kcrez
41072 bc(k,ld)=rezc(ld)
do 41073 ld=1,kdrez
41073 bd(k,ld)=rezd(ld)
c write(5,15684) b(k)
15684 format(1x,'b(k) после=',d30.16)

k=k+1
if(k.le.n) goto 71
i=i+1
if(i.lt.n) goto 81
c pause' после prjm'
c write(5,*) 'матрица после прямого хода'
do 13451 i=1,n
do 13458 j=1,n
c write(5,13452) i,j
c write(*,*) i,j
c pause
13452 format(/1x,'i=',i3,' j=',i3)
c write(5,13453) (a(i,j),j=1,n)
zn=znak(i,j)
ncr=nc(i,j)
ndr=nd(i,j)
c write(5,1371) zn
c write(5,1892) ncr,ndr
c write(*,*) zn,ncr,ndr
c pause
1371 format(1x,'знак=',i2)
c write(5,1372) (ac(i,j,l),l=1,ncr)
1372 format(1x,'целая часть=',60i1)
c write(5,1373) (ad(i,j,l),l=1,ndr)
1373 format(1x,'дробная часть=',60i1/)
13453 format(1x,4d18.12)
13458 continue

```

```
13451 continue
c   write(5,*) 'диагональ матрицы после прямого хода'
c   do 56 i=1,n
c   write(5,57) i,a(i,i)
57   format(1x,'i=',i5,' a=',f36.18)
56   continue
c   write(5,*) ' свободные члены после прямого хода'
c   do 1558 i=1,n
c   write(5,1557) i
1557   format(/1x,'i=',i3)
      ncr=ncb(i)
      ndr=ncd(i)
      zn=znp(i)
c   write(5,1371) zn
c   write(5,1372) (bc(i,l),l=1,ncr)
c   write(5,1373) (bd(i,l),l=1,ndr)
1558   continue
c   stop
c   pause 'pered obrdl'
      call obrdl(n,znak,nc,nd,ac,ad,znp,ncb,ncd,bc,bd,znx,nxc,nxd,
*xc,xd,igau,mnper,anc,andr,krazr)
c   pause 'posle obrdl'
      n=nis
c   do 7861 i=1,n
c   write(6,7862) i
7862   format(1x,'в гау неизвестное,номер=',i3)
c   n1c=nxc(i)
c   n1d=nxd(i)
c   write(6,1371) znx(i)
c   write(6,1892) n1c,n1d
c   write(6,1372) (xc(i,ld),ld=1,n1c)
c   write(6,1373) (xd(i,ld),ld=1,n1d)
7861   continue
      return
      end
```

**Набор nimax.for**

```

$large
  subroutine nimax(i,j,n,nc,nd,znak,ac,ad,nbc,nbd,znp,bc,bd,
  *mnp, krazr)
  dimension mnp(100)
  integer*2 znak(31,31),nc(31,31),nd(31,31),tc(400),td(400),
  *ac(31,31,20),ad(31,31,20),znp(31),nbc(31),nbd(31)
  integer*2 bc(31,20),bd(31,20)
  integer*2 xmaxc(400),xmaxd(400)
c   integer znak,ac,ad,znp,bc,bd
  integer xmax

c-нахождение максимального элемента
c   xmax=dabs(a(i,j))
  ncel=nc(i,j)
  ndr=nd(i,j)
  zmax=1
  kcmx=ncel
  kdmax=ndr
  do 41001 ld=1,ncel
  xmaxc(ld)=ac(i,j,ld)
41001 continue
  do 41002 ld=1,ndr
41002 xmaxd(ld)=ad(i,j,ld)

  imax=i
  jmax=j
  do 38250 ipoi=i,n
  do 38250 jpoi=j,n
c   if(dabs(a(ipoi,jpoi)).le.xmax) goto 38250
c   xmax=dabs(a(ipoi,jpoi))
  ncr=nc(ipoi,jpoi)
  ndr=nd(ipoi,jpoi)
  if(ncr.lt.kcmx) goto 38250
  if(ncr.gt.kcmx) goto 41003
  do 41004 ld=1,ncr
  if(ac(ipoi,jpoi,ld).gt.xmaxc(ld)) goto 41003
  if(ac(ipoi,jpoi,ld).lt.xmaxc(ld)) goto 38250
41004 continue
  kmin=kdmax
  if(kmin.gt.ndr) kmin=ndr
  do 41005 ld=1,kmin
  if(ad(ipoi,jpoi,ld).gt.xmaxd(ld)) goto 41003
  if(ad(ipoi,jpoi,ld).lt.xmaxd(ld)) goto 38250
41005 continue
  if(kdmax.ge.ndr) goto 38250
41003 continue

  imax=ipoi
  jmax=jpoi
  kcmx=ncr
  kdmax=ndr
  do 41006 ld=1,ncr
  xmaxc(ld)=ac(ipoi,jpoi,ld)
41006 continue
  do 41007 ld=1,ndr
  xmaxd(ld)=ad(ipoi,jpoi,ld)
41007 continue
38250 continue
c-пересылка строки i со строкой imax
  do 38251 jper=j,n
c   t=a(i,jper)

```

```

c      a(i,jper)=a(imax,jper)
c      a(imax,jper)=t
      ktc=nc(i,jper)
      ktd=nd(i,jper)
      do 41008 ld=1,ktc
41008  tc(ld)=ac(i,jper,ld)
      continue
      do 41009 ld=1,ktd
      td(ld)=ad(i,jper,ld)
41009  continue
      znt=znak(i,jper)
      ncr=nc(imax,jper)
      ndr=nd(imax,jper)
      do 41010 ld=1,ncr
      ac(i,jper,ld)=ac(imax,jper,ld)
41010  continue
      do 41011 ld=1,ndr
41011  ad(i,jper,ld)=ad(imax,jper,ld)
      znak(i,jper)=znak(imax,jper)
      nc(i,jper)=ncr
      nd(i,jper)=ndr
      do 41012 ld=1,ktc
      ac(imax,jper,ld)=tc(ld)
41012  continue
      do 41013 ld=1,ktd
      ad(imax,jper,ld)=td(ld)
41013  continue
      znak(imax,jper)=znt
      nc(imax,jper)=ktc
      nd(imax,jper)=ktd
38251  continue
      ncr=ncr(i)
      ndr=ndr(i)
      iznr=iznr(i)
      do 41014 ld=1,ncr
      tc(ld)=bc(i,ld)
41014  continue
      do 41015 ld=1,ndr
      td(ld)=bd(i,ld)
41015  continue
      nbc(i)=nbc(imax)
      nbd(i)=nbd(imax)
      znp(i)=znp(imax)
      n1c=nbc(imax)
      n1d=nbd(imax)
      do 41016 ld=1,n1c
      bc(i,ld)=bc(imax,ld)
41016  continue
      do 41017 ld=1,n1d
      bd(i,ld)=bd(imax,ld)
41017  continue
      nbc(imax)=ncr
      nbd(imax)=ndr
      znp(imax)=iznr
      do 41018 ld=1,ncr
      bc(imax,ld)=tc(ld)
41018  continue
      do 41019 ld=1,ndr
      bd(imax,ld)=td(ld)
41019  continue

c      t=b(i)
c      b(i)=b(imax)

```

```

c      b(imax)=t
c-пересылка столбца j со столбцом jmax
      do 38252 ipoi=1,n
c      t=a(ipoi,j)
c      a(ipoi,j)=a(ipoi,jmax)
c      a(ipoi,jmax)=t
      ncr=nc(ipoi,j)
      ndr=nd(ipoi,j)
      iznr=znak(ipoi,j)
      do 41020 ld=1,ncr
41020  tc(ld)=ac(ipoi,j,ld)
      do 41021 ld=1,ndr
41021  td(ld)=ad(ipoi,j,ld)
      nc(ipoi,j)=nc(ipoi,jmax)
      nd(ipoi,j)=nd(ipoi,jmax)
      znak(ipoi,j)=znak(ipoi,jmax)
      nlc=nc(ipoi,jmax)
      nld=nd(ipoi,jmax)
      do 41022 ld=1,nlc
41022  ac(ipoi,j,ld)=ac(ipoi,jmax,ld)
      do 41023 ld=1,nld
41023  ad(ipoi,j,ld)=ad(ipoi,jmax,ld)
      nc(ipoi,jmax)=ncr
      nd(ipoi,jmax)=ndr
      znak(ipoi,jmax)=iznr
      do 41024 ld=1,ncr
41024  ac(ipoi,jmax,ld)=tc(ld)
      do 41025 ld=1,ndr
41025  ad(ipoi,jmax,ld)=td(ld)
38252  continue
c-пересылка в массиве номеров mpre столбцов
      jpre=mpre(j)
      mpre(j)=mpre(jmax)
      mpre(jmax)=jpre
      return
      end

```

## Набор obrdl.for

\$large

c-обратный ход

c-метод Гаусса решения систем с выбором главного элемента, длинная арифметика

c-ac(i,j,l)-целая часть элемента i,j матрицы системы

c-ad(I,j,l)-дробная часть элемента I,j матрицы системы

c-nc(I,j)-количество разрядов целой части элемента I,j

c-nd(I,j)-количество разрядов дробной части элемента I,j

c-znak(I,j)-знак элемента I,j

c-bc(I,l)-целая часть правой части i-ого уравнения

c-bd(I,l)-дробная часть правой части i-го уравнения

c-nbc(i)-длина целой части правой части i-го уравнения

c-nbd(i)-длина дробной части правой части i-го уравнения

c-znp(i)-знак правой части i-ого уравнения

c-xc(I,l)-целая часть i-го неизвестного

c-xd(I,l)-дробная часть i-го неизвестного

c-nxc(i)-длина целой части i-ого неизвестного

c-nxd(i)-длина дробной части i-ого неизвестного

c-znx(i)-знак i-ой неизвестной

```
subroutine obrdl(n,znak,nc,nd,ac,ad,znp,nbc,nbd,bc,bd,znx,nxc,nxd,
```

```
*xc,xd,igau,mnper,anc,andr,krazr)
```

```
dimension mnper(100)
```

```
integer*2 znak(31,31),nc(31,31),nd(31,31),tc(400),td(400),
```

```
*ac(31,31,20),ad(31,31,20),znp(31),nbc(31),nbd(31)
```

```
integer*2 bc(31,20),bd(31,20),znx(31),nxc(31),nxd(31)
```

```
integer*2 xc(31,20),xd(31,20)
```

```
integer*2 rezc(400),rezd(400),anc(400),andr(400),
```

```
*icsum(400),idsun(400)
```

```
integer*2 rc(400),rd(400)
```

```
c integer znak,ac,ad,znp,bc,bd,znx,xc,xd,rezc,rezd,rc,rd
```

```
c integer tc,td,anc,andr
```

```
c pause 'obr d start'
```

```
nis=n
```

```
c if(dabs(a(n,n)).gt.0.1d-38) goto 82
```

```
nct=nc(n,n)
```

```
ndt=nd(n,n)
```

```
izt=znak(n,n)
```

```
do 41074 ld=1,nct
```

```
41074 tc(ld)=ac(n,n,ld)
```

```
do 41075 ld=1,ndt
```

```
41075 td(ld)=ad(n,n,ld)
```

```
do 41076 ld=1,nct
```

```
if(tc(ld).ne.anc(ld)) goto 82
```

```
41076 continue
```

```
do 41077 ld=1,ndt
```

```
if(td(ld).ne.andr(ld)) goto 82
```

```
41077 continue
```

```
write(6,*) 'матрица вырожденная'
```

```
c pause 'obr wir'
```

```
igau=0
```

```
return
```

```
82 continue
```

```
1371 format(1x,'знак=',i2)
```

```
1892 format(1x,'длина целой части=',i3,'длина дробной части=',i3)
```

```
1372 format(1x,'целая часть=',60i1)
```

```
1373 format(1x,'дробная часть=',60i1)
```

```
c pause 'obr 1'
```

```
nt=n
```

```
c x(n)=b(n)/a(n,n)
```

```
nrc=nbc(n)
```

```

      ndr=nbdc(n)

      izr=znp(n)
      do 41088 ld=1,ncr
41088 rc(ld)=bc(n,ld)
      do 41089 ld=1,ndr
41089 rd(ld)=bd(n,ld)
      c      pause 'obr 41089'
      c      write(6,*) '41089 do delzn'
      c      write(6,*) 'Обратный ход,b(n)/a(n,n)'
      c      write(6,5621) n
5621 format(1x,'n=',i3)
      c      write(6,*) 'делимое b(n)'
      c      write(6,1371) izr
      c      write(6,1892) ncr,ndr
      c      write(6,1372) (rc(ld),ld=1,ncr)
      c      write(6,1373) (rd(ld),ld=1,ndr)
      c      write(6,*) 'делитель a(n,n)'
      c      write(6,1371) izt
      c      write(6,1892) nct,ndt
      c      write(6,1372) (tc(ld),ld=1,nct)
      c      write(6,1373) (td(ld),ld=1,ndt)
      c      pause 'obr delzn pered'
      call delzn(izr,izt,izrez,ncr,ndr,rc,rd,nct,ndt,tc,td,
      *kcrez,kdrez,rezc,rezd,krazr)
      c      pause 'obr delzn после'
      c      write(6,*) 'частное x(n)=b(n)/a(n,n)'
      c      write(6,5621) n
      c      write(6,1371) izrez
      c      write(6,1892) kcrez,kdrez
      c      write(6,1372) (rezc(ld),ld=1,kcrez)
      c      write(6,1373) (rezd(ld),ld=1,kdrez)
      c      write(5,*) '41089 после delzn'
      nxc(n)=kcrez
      nxd(n)=kdrez
      znx(n)=izrez
      do 41090 ld=1,kcrez
41090 xc(n,ld)=rezc(ld)
      do 41091 ld=1,kdrez
41091 xd(n,ld)=rezd(ld)
      c      write(6,86) x(n)
85      n=n-1
      c      write(*,*) n
      c      pause 'obr 2'
      if(n.lt.1) goto 83
      c      sum=0
      ncsun=1
      ndsum=1
      izsum=1
      do 41092 ld=1,ncsum
      icsum(ld)=0
      idsum(ld)=0
41092 continue
      j=n+1
84      continue
      ncr=nc(n,j)
      ndr=nd(n,j)
      izr=znak(n,j)
      do 41093 ld=1,ncr
41093 rc(ld)=ac(n,j,ld)
      do 41094 ld=1,ndr
41094 rd(ld)=ad(n,j,ld)
      nc1=nxc(j)

```

```

        nd1=nxd(j)
        iz1=znx(j)
        do 41095 ld=1,nc1
41095 tc(ld)=xc(j,ld)
        do 41096 ld=1,nd1
41096 td(ld)=xd(j,ld)
c      write(6,*) 'умножение a(n,j)*x(j) '
c      write(6,5622) n,j
5622 format(1x,'n=',i3,' j=',i3)
c      write(6,*) 'множитель a(n,j) '
c      write(6,1371) izr
c      write(6,1892) ncr,ndr
c      write(6,1372) (rc(ld),ld=1,ncr)
c      write(6,1373) (rd(ld),ld=1,ndr)
c      write(6,*) 'множитель x(j) '
c      write(6,1371) iz1
c      write(6,1892) ncl,ndl
c      write(6,1372) (tc(ld),ld=1,ncl)
c      write(6,1373) (td(ld),ld=1,ndl)
        call ymnzn(izr,iz1,izrez,ncr,ndr,rc,rd,ncl,ndl,tc,td,
        *kcrez,kdrez,rezc,rezd,krazr)
c      write(6,*) 'произведение a(n,j)*x(j) '
c      write(6,5622) n,j
c      write(6,1371) izrez
c      write(6,1892) kcrez,kdrez
c      write(6,1372) (rezc(ld),ld=1,kcrez)
c      write(6,1373) (rezd(ld),ld=1,kdrez)
        ncr=kcrez
        ndr=kdrez
        izr=izrez
        do 41097 ld=1,ncr
41097 rc(ld)=rezc(ld)
        do 41098 ld=1,ndr
41098 rd(ld)=rezd(ld)
c      write(6,*) 'сумма sum+a(n,j)*x(j) '
c      write(6,5622) n,j
c      write(6,*) 'слагаемое sum'
c      write(6,1371) izsum
c      write(6,1892) ncsun,ndsum
c      write(6,1372) (icsun(ld),ld=1,ncsun)
c      write(6,1373) (idsun(ld),ld=1,ndsum)
c      write(6,*) 'слагаемое a(n,j)*x(j) '
c      write(6,5622) n,j
c      write(6,1371) izr
c      write(6,1892) ncr,ndr
c      write(6,1372) (rc(ld),ld=1,ncr)
c      write(6,1373) (rd(ld),ld=1,ndr)
        call slzn(izsum,izr,izrez,ncsun,ndsum,icsun,idsun,ncr,ndr,
        *rc,rd,kcrez,kdrez,rezc,rezd,krazr)
c      write(6,*) 'сумма sum+a(n,j)*x(j) '
c      write(6,5622) n,j
c      write(6,1371) izrez
c      write(6,1892) kcrez,kdrez
c      write(6,1372) (rezc(ld),ld=1,kcrez)
c      write(6,1373) (rezd(ld),ld=1,kdrez)
        ncsun=kcrez
        ndsum=kdrez
        izsum=izrez
        do 41099 ld=1,kcrez
41099 icsun(ld)=rezc(ld)
        do 41100 ld=1,kdrez
41100 idsun(ld)=rezd(ld)
c84  sum=sum+a(n,j)*x(j)

```

```

      j=j+1
c     write(*,*) j
c     pause 'obr j'
      if(j.le.nt) goto 84
c     x(n)=b(n)-sum
      ncr=ncb(n)
      ndr=ncd(n)
      izr=znp(n)
      do 41101 ld=1,ncr
41101 rc(ld)=bc(n,ld)
      do 41102 ld=1,ndr
41102 rd(ld)=bd(n,ld)
c     write(5,*) '41102 do wiczn'
c     write(6,*) 'уменьшаемое b(n) '
c     write(6,5621) n
c     write(6,1371) izr
c       write(6,1892) ncr,ndr
c       write(6,1372) (rc(1),l=1,ncr)
c       write(6,1373) (rd(1),l=1,ndr)
c     write(6,*) 'вычитаемое sum'
c     write(6,1371) izsum
c       write(6,1892) ncsum,ndsum
c       write(6,1372) (icsum(1),l=1,ncsum)
c       write(6,1373) (idsum(1),l=1,ndsum)
      call wiczn(izr,izsum,izrez,ncr,ndr,rc,rd,ncsum,ndsum,icsum,
*idsum,kcrez,kdrez,rezc,rezd,krazr)
c     write(6,*) 'разность x(n)=b(n)-sum'
c     write(6,3201) n
3201 format(1x,'неизвестное n=',i3)
c     write(6,1371) izrez
c     write(6,1892) kcrez,kdrez
c     write(6,1372) (rezc(ld),ld=1,kcrez)
c     write(6,1373) (rezd(ld),ld=1,kdrez)
c     write(5,*) '41102 после wiczn'
      nxc(n)=kcrez
      nxd(n)=kdrez
      znx(n)=izrez
      do 41103 ld=1,kcrez
41103 xc(n,ld)=rezc(ld)
      do 41104 ld=1,kdrez
41104 xd(n,ld)=rezd(ld)
c     write(6,86) x(n)
      goto 85
83     continue
c     pause 'obr 83'
c     write(5,86) (x(i),i=1,nt)
86     format(1x,5f15.5)
      igau=1
c-расстановка значений в векторе решения X
      do 38254 i=1,nt
        j=mnper(i)
c     b(j)=x(i)
        nbc(j)=nxc(i)
        nbd(j)=nxd(i)
        znp(j)=znx(i)
        n1c=nxc(i)
        n1d=nxd(i)
        do 41105 ld=1,n1c
41105 bc(j,ld)=xc(i,ld)
        do 41106 ld=1,n1d
41106 bd(j,ld)=xd(i,ld)
38254 continue
      do 38255 i=1,nt

```

```
c      x(i)=b(i)
      nxc(i)=nbc(i)
      nxd(i)=nbd(i)
      znx(i)=znp(i)
      n1c=nbc(i)
      n1d=nbd(i)
      do 41107 ld=1,n1c
41107  xc(i,ld)=bc(i,ld)
      do 41108 ld=1,n1d
41108  xd(i,ld)=bd(i,ld)
c      write(6,7821) i
7821  format(1x,'обр. неизвестное,номер=',i3)
c      write(6,1371) znx(i)
c      write(6,1892) nxc(i),nxd(i)
c      write(6,1372) (xc(i,ld),ld=1,n1c)
c      write(6,1373) (xd(i,ld),ld=1,n1d)
38255 continue
      n=nis
      return
      end
```

## Набор prowg.for

```

$large
      subroutine prowg(n,znak,nc,nd,ac,ad,
      *znp,nbc,nbd,bc,bd,znx,nxc,nxd,xc,xd,igau,krazr)
с-проверка метода Гаусса
с-ac(i,j,l)-целая часть элемента i,j матрицы системы
с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
с-nc(I,j)-количество разрядов целой части элемента I,j
с-nd(I,j)-количество разрядов дробной части элемента I,j
с-znak(I,j)-знак элемента I,j
с-bc(I,l)-целая часть правой части i-ого уравнения
с-bd(I,l)-дробная часть правой части i-го уравнения
с-nbc(i)-длина целой части правой части i-го уравнения
с-nbd(i)-длина дробной части правой части i-го уравнения
с-znp(i)-знак правой части i-ого уравнения
с-xc(I,l)-целая часть i-го неизвестного
с-xd(I,l)-дробная часть i-го неизвестного
с-nxc(i)-длина целой части i-ого неизвестного
с-nxd(i)-длина дробной части i-ого неизвестного
с-znx(i)-знак i-ой неизвестной
      integer*2 znak(31,31),nc(31,31),nd(31,31),
      *ac(31,31,20),ad(31,31,20),znp(31),nbc(31),nbd(31)
      integer*2 bc(31,20),bd(31,20),znx(31),nxc(31),nxd(31)
      integer*2 xc(31,20),xd(31,20),icel(400),idd(400)
      integer*2 icl(400),idl(400)
      integer*2 ic2(400),id2(400),icrez(400),idrez(400)
      integer*2 icsum(400),idsum(400)
с double precision c,b,x,sum
      integer zn
      open(9,file='promprow',status='old')
с open(5,file='prowgadl',status='Unknown')

      read(9,*) n,krazr
с write(*,*) n,krazr
с pause '1'
      write(5,*) 'проверка решения,матрица системы'
      do 1 i=1,n
      do 2 j=1,n
      write(5,44557) i,j
44557 format(1x,'элемент c(',i3,',',i3,')')
      read(9,250) znak(i,j),ncz,ndz,(ac(i,j,l),l=1,ncz),
      *(ad(i,j,l),l=1,ndz)
      write(5,26) znak(i,j)
      write(5,27) ncz,ndz
      write(5,28) (ac(i,j,ld),ld=1,ncz)
      write(5,29) (ad(i,j,ld),ld=1,ndz)

250 format(3i2,80i1)
с write(5,251) znak(i,j),ncz,ndz,(ac(i,j,l),l=1,ncz),
с *(ad(i,j,l),l=1,ndz)
251 format(1x,3i3,60i1)
      nc(i,j)=ncz
      nd(i,j)=ndz

2 continue
1 continue
      write(5,*) 'свободные члены'
      do 3 i=1,n
      write(5,44558) i
44558 format(1x,'i=',i3)
      read(9,250) znp(i),nbcz,nbdz,(bc(i,l),l=1,nbcz),
      *(bd(i,l),l=1,nbdz)
      write(5,26) znp(i)

```

```

write(5,27) nbcz,nbdz
write(5,28) (bc(i,ld),ld=1,nbcz)
write(5,29) (bd(i,ld),ld=1,nbdz)

c write(6,251) znp(i),nbcz,nbdz,(bc(i,l),l=1,nbcz),
c *(bd(i,l),l=1,nbdz)
nbc(i)=nbcz
nbd(i)=nbdz
3 continue
write(5,*) 'неизвестные системы'
do 3001 i=1,n
write(5,44558) i
c read(4,250) znx(i),nbcz,nbdz,(xc(i,l),l=1,nbcz),
c *(xd(i,l),l=1,nbdz)
nbcz=nxc(i)
nbdz=nxd(i)
write(5,26) znx(i)
write(5,27) nbcz,nbdz
write(5,28) (xc(i,ld),ld=1,nbcz)
write(5,29) (xd(i,ld),ld=1,nbdz)

c write(6,251) znx(i),nbcz,nbdz,(xc(i,l),l=1,nbcz),
c *(xd(i,l),l=1,nbdz)
c nxc(i)=nbcz
c nxd(i)=nbdz
3001 continue

do 231 i=1,n
c sum=0
isc=1
isd=1
icsum(1)=0
idsum(1)=0
do 300 j=1,n
izn2=znx(j)
ikc2=nxc(j)
ikd2=nxd(j)
do 801 ld=1,ikc2
801 ic2(ld)=xc(j,ld)
do 802 ld=1,ikd2
802 id2(ld)=xd(j,ld)
c write(6,451) i,j
451 format(/1x,'i=',i3,' j=',i3)
izn1=znak(i,j)
ikc1=nc(i,j)
ikd1=nd(i,j)
do 701 ld=1,ikc1
701 ic1(ld)=ac(i,j,ld)
do 702 ld=1,ikd1
702 id1(ld)=ad(i,j,ld)
c write(6,*) 'элемент c(i,j)'
c write(6,26) izn1
c26 format(1x,'знак=',i3)
c write(6,27) ikc1,ikd1
c27 format(1x,'длина целой=',i3,' длина дробной=',i3)
c write(6,28) (ic1(ld),ld=1,ikc1)
c28 format(1x,'целая часть=',50i1)
c write(6,29) (id1(ld),ld=1,ikd1)
c29 format(1x,'дробная часть=',50i1)
c write(6,*) 'множитель x(j)'
c write(6,26) izn2
c write(6,27) ikc2,ikd2
c write(6,28) (ic2(ld),ld=1,ikc2)

```

```

c      write(6,29) (id2(ld),ld=1,ikd2)

      call ymnzn(izn1,izn2,izrez,ikc1,ikd1,ic1,id1,ikc2,ikd2,ic2,id2,
*kcvez,kdvez,icvez,idvez,krazr)
c      write(6,*) 'произведение с(i,j)*x(j) '
c      write(6,26) izvez
c      write(6,27) kcvez,kdvez
c      write(6,28) (icvez(ld),ld=1,kcvez)
c      write(6,29) (idvez(ld),ld=1,kdvez)

      izn2=izvez
      ikc2=kcvez
      ikd2=kdvez
      do 323 ld=1,kcvez
323    ic2(ld)=icvez(ld)
      do 423 ld=1,kdvez
423    id2(ld)=idvez(ld)
c      write(6,*) 'слагаемое sum'
c      write(6,26) izsum
c26    format(1x,'знак=',i3)
c      write(6,27) isc,isd
c27    format(1x,'длина целой=',i3,' длина дробной=',i3)
c      write(6,28) (icsum(ld),ld=1,isc)
c28    format(1x,'целая часть=',50i1)
c      write(6,29) (idsum(ld),ld=1,isd)
c29    format(1x,'дробная часть=',50i1)
c      write(6,*) 'второе слагаемое с(i,j)*x(j) '
c      write(6,26) izn2
c      write(6,27) ikc2,ikd2
c      write(6,28) (ic2(ld),ld=1,ikc2)
c      write(6,29) (id2(ld),ld=1,ikd2)

      call slzn(izsum,izn2,izvez,isc,isd,icsum,idsum,ikc2,ikd2,
*ic2,id2,kcvez,kdvez,icvez,idvez,krazr)
c      write(6,*) 'сумма sum+c(i,j)*x(j) '
c      write(6,26) izvez
c      write(6,27) kcvez,kdvez
c      write(6,28) (icvez(ld),ld=1,kcvez)
c      write(6,29) (idvez(ld),ld=1,kdvez)

      izsum=izvez
      isc=kcvez
      isd=kdvez
      do 555 ld=1,kcvez
555    icsum(ld)=icvez(ld)
      do 666 ld=1,kdvez
666    idsum(ld)=idvez(ld)
300    continue
      izb=znp(i)
      icb=ncb(i)
      idb=ncd(i)
      do 815 ld=1,icb
815    ic1(ld)=bc(i,ld)
      do 816 ld=1,idb
816    id1(ld)=bd(i,ld)
c      write(6,*) 'уменьшаемое sum'
c      write(6,26) izsum
c26    format(1x,'знак=',i3)
c      write(6,27) isc,isd
c27    format(1x,'длина целой=',i3,' длина дробной=',i3)
c      write(6,28) (icsum(ld),ld=1,isc)
c28    format(1x,'целая часть=',50i1)
c      write(6,29) (idsum(ld),ld=1,isd)

```

```

c29  format(1x,'дробная часть=',50i1)
c    write(6,*) 'вычитаемое b(i)'
c    write(6,26) izb
c    write(6,27) icb,idb
c    write(6,28) (ic1(ld),ld=1,icb)
c    write(6,29) (id1(ld),ld=1,idb)

      call wiczn(izsum,izb,izrez,isc,isd,icsum,idsum,icb,idb,ic1,id1,
      *kcrez,kdrez,icrez,idrez,krazr)
c    write(6,*) 'разница sum-b(i)'
c    write(6,26) izrez
c    write(6,27) kcrez,kdrez
c    write(6,28) (icrez(ld),ld=1,kcrez)
c    write(6,29) (idrez(ld),ld=1,kdrez)

      write(5,25) i
25   format(1x,'уравнение-',i3)
      write(5,*) 'правая часть,исход'
      write(5,26) izb
26   format(1x,'знак=',i3)
      write(5,27) icb,idb
27   format(1x,'длина целой=',i3,' длина дробной=',i3)
      write(5,28) (ic1(ld),ld=1,icb)
28   format(1x,'целая часть=',50i1)
      write(5,29) (id1(ld),ld=1,idb)
29   format(1x,'дробная часть=',50i1)
      write(5,*) 'правая часть,счет'
      write(5,26) izsum
      write(5,27) isc,isd
      write(5,28) (icsum(ld),ld=1,isc)
      write(5,29) (idsum(ld),ld=1,isd)
      write(5,*) 'разница'
      write(5,26) izrez
      write(5,27) kcrez,kdrez
      write(5,28) (icrez(ld),ld=1,kcrez)
      write(5,29) (idrez(ld),ld=1,kdrez)
231  continue
c    close(4,status='keep')
c    close(6,status='keep')
      close(9,status='keep')
      return
      end

```

## Набор sl.for

c-длинное сложение

\$large

```

subroutine sl(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
integer*2 a(400),b(400),c(400),ad(400),bd(400),cd(400)
integer*2 rezc(400),rezdr(400)
integer rez,os,p
c   open(5,file='summod',status='Unknown')
c   write(*,111)
111  format(1x,'введите длину целой части первого слагаемого')
c   read(*,*) n
c   write(*,112)
112  format(1x,'введите длину дробной части первого слагаемого')
c   read(*,*) nd
c   write(*,1111)
1111 format(1x,'введите длину целой части второго слагаемого')
c   read(*,*) m
c   write(*,1112)
1112 format(1x,'введите длину дробной части второго слагаемого')
c   read(*,*) md

c   write(*,334)
334  format(1x,'введите через пробел разряды целой части',
*' первого слагаемого')
c   read(*,*) (a(i),i=1,n)
c   write(*,335)
335  format(1x,'введите через пробел разряды дробной части',
*' первого слагаемого')

c   read(*,*) (ad(i),i=1,nd)
c   write(*,1334)
1334 format(1x,'введите через пробел разряды целой части',
*' второго слагаемого')

c   read(*,*) (b(i),i=1,m)
c   write(*,1335)
1335 format(1x,'введите через пробел разряды дробной части',
*' второго слагаемого')

c   read(*,*) (bd(i),i=1,md)
c   write(5,*) 'целая часть первого слагаемого'
c   write(5,445) (a(i),i=1,n)
445  format(1x,70i1)
c   write(5,*) 'дробная часть первого слагаемого'
c   write(5,445) (ad(i),i=1,nd)
c   write(5,*) 'целая часть второго слагаемого'
c   write(5,445) (b(i),i=1,m)
c   write(5,*) 'дробная часть второго слагаемого'
c   write(5,445) (bd(i),i=1,md)
c   if(krazr.le.100) goto 66791
c   write(*,*) 'сложение, количество разрядов>100!'
c   pause
c   stop
66791 continue
      do 66875 i=1,400
        rezc(i)=0
66875 rezdr(i)=0
      if(n.ge.m) goto 897
      do 678 i=1,n
        c(i)=a(i)

```

```

678  continue
      do 679 i=1,nd
        cd(i)=ad(i)
679  continue
      do 680 i=1,m
        a(i)=b(i)
680  continue
      do 681 i=1,md
        ad(i)=bd(i)
681  continue
      do 682 i=1,n
        b(i)=c(i)
682  continue
      do 683 i=1,nd
        bd(i)=cd(i)
683  continue
      ir=n
      n=m
      m=ir
      ir=nd
      nd=md
      md=ir
897  continue
с-формирование двух массивов для сложения
      max=nd
      if(max.lt.md) max=md
      if(md.eq.max) goto 78
      in=md+1
      do 79 i=in,max
        bd(i)=0
79  continue
      goto 81
78  continue
      in=nd+1
      do 82 i=in,max
        ad(i)=0
82  continue
81  continue
      it=0
      in=n+1
      ik=n+max
      do 100 i=in,ik
        it=it+1
        a(i)=ad(it)
100  continue
      it=0
      in=m+1
      ik=m+max
      do 101 i=in,ik
        it=it+1
        b(i)=bd(it)
101  continue

с-сложение
      i=0
      p=0
      nt=n+max
      mt=m+max
7  continue
      if(mt.lt.1) rez=a(nt)
      if(mt.ge.1) rez=a(nt)+b(mt)
      rez=rez+p
      il=rez/10

```

```

os=rez-i1*10
i=i+1
c(i)=os
c  if(i.ne.1) goto 1
p=i1
nt=nt-1
mt=mt-1
if((p.eq.0).and.(mt.lt.1)) goto 4
if((nt.lt.1).and.(p.ne.0)) goto 5
if((nt.lt.1).and.(p.eq.0)) goto 4
goto 7
5  continue
i=i+1
c(i)=p
goto 67
4  continue

89  continue
if(nt.lt.1) goto 67
i=i+1
c(i)=a(nt)
nt=nt-1
goto 89
67  continue
c  write(5,*) 'целая часть суммы'
ik=imax+1
c  write(5,445) (c(j),j=i,ik,-1)
c  write(5,*) 'дробная часть суммы'
c  write(5,445) (c(j),j=imax,1,-1)

kcel=i-imax
kdro=imax
it=0
do 1121 j=i,ik,-1
it=it+1
rezc(it)=c(j)
1121 continue
it=0
do 1122 j=imax,1,-1

it=it+1
rezdr(it)=c(j)
1122 continue
if(kcel.ne.0) goto 971
kcel=1
rezc(1)=0
971 continue
if(kdro.ne.0) goto 972
kdro=1
rezdr(1)=0
972 continue
if(kdro.gt.krazr) kdro=krazr
c  if(kcel.le.krazr) goto 67291
c  write(*,67292) kcel,krazr
67292 format(1x,'сложение, количество разрядов в целой части=',i3,
*' больше заданного=',i3)
c  pause
c  stop
67291 continue
return
c  close(5,status='keep')
end

```

## Набор slzn.for

```

$large
  subroutine slzn(za,zb,zc,n,nd,a,ad,m,md,b,bd,
    *kcel,kdro,rezc,rezdr,krazr)
с-длинное сложение с учетом знаков слагаемых
  integer*2 a(400),b(400),c(400),ad(400),bd(400),cd(400)
  integer*2 rezc(400),rezdr(400),ar(400),adr(400)
  integer*2 ai(400),adi(400),bi(400),bdi(400)
  integer rez,os,p,za,zb,zc
  integer zais,zbis
с
  open(5,file='dlsumzn',status='Unknown')
с
  write(*,2001)
2001 format(1x,'введите знак первого слагаемого,1-плюс,-1-минус')
с
  read(*,*) za
с
  write(*,2002)
2002 format(1x,'введите знак второго слагаемого,1-плюс,-1-минус')
с
  read(*,*) zb
с
  write(*,111)
111 format(1x,'введите длину целой части первого слагаемого')
с
  read(*,*) n
с
  write(*,112)
112 format(1x,'введите длину дробной части первого слагаемого')
с
  read(*,*) nd
с
  write(*,1111)
1111 format(1x,'введите длину целой части второго слагаемого')
с
  read(*,*) m
с
  write(*,1112)
1112 format(1x,'введите длину дробной части второго слагаемого')
с
  read(*,*) md

с
  write(*,334)
334 format(1x,'введите через пробел разряды целой части',
  *' первого слагаемого')
с
  read(*,*) (a(i),i=1,n)
с
  write(*,335)
335 format(1x,'введите через пробел разряды дробной части',
  *' первого слагаемого')

с
  read(*,*) (ad(i),i=1,nd)
с
  write(*,1334)

1334 format(1x,'введите через пробел разряды целой части',
  *' второго слагаемого')

с
  read(*,*) (b(i),i=1,m)
с
  write(*,1335)
1335 format(1x,'введите через пробел разряды дробной части',
  *' второго слагаемого')

с
  read(*,*) (bd(i),i=1,md)
с
  write(5,*) 'целая часть первого слагаемого'
с
  write(5,445) (a(i),i=1,n)
445 format(1x,70i1)
с
  write(5,*) 'дробная часть первого слагаемого'
с
  write(5,445) (ad(i),i=1,nd)
с
  write(5,*) 'целая часть второго слагаемого'
с
  write(5,445) (b(i),i=1,m)
с
  write(5,*) 'дробная часть второго слагаемого'
с
  write(5,445) (bd(i),i=1,md)
  zais=za

```

```

zbis=zb
do 5671 i=1,n
ai(i)=a(i)
5671 continue
do 5672 i=1,nd
adi(i)=ad(i)
5672 continue
do 5673 i=1,m
bi(i)=b(i)
5673 continue
do 5674 i=1,md
bdi(i)=bd(i)
5674 continue
ni=n
ndi=nd
mi=m
mdi=md

ip=za*zb
if(ip.gt.0) goto 781
if(za.eq.1) goto 782
1901 if(m.gt.n) goto 783
if(m.eq.n) goto 784
call wic(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
zc=za
goto 785
784 continue
do 786 i=1,m
if(a(i).gt.b(i)) goto 787
if(a(i).lt.b(i)) goto 788
786 continue
nmin=nd
if(nmin.gt.md) nmin=md
do 791 i=1,nmin
if(ad(i).gt.bd(i)) goto 792
if(ad(i).lt.bd(i)) goto 793
791 continue
if(nd.le.md) goto 795
call wic(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
zc=za
goto 785
795 continue
call wic(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
zc=zb
goto 785
793 continue
call wic(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
zc=zb
goto 785
792 continue
call wic(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
zc=za
goto 785
788 continue
call wic(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
zc=zb
goto 785
787 continue
call wic(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
zc=za
goto 785
783 continue
call wic(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)

```

```

      zc=zb
      goto 785
782  continue
      do 1782 i=1,n
      ar(i)=a(i)
1782  continue
      do 1783 i=1,nd
      adr(i)=ad(i)
1783  continue
      nr=n
      ndr=nd
      do 1784 i=1,m
      a(i)=b(i)
1784  continue
      do 1785 i=1,md
      ad(i)=bd(i)
1785  continue
      n=m
      nd=md
      m=nr
      md=ndr
      do 1786 i=1,nr
      b(i)=ar(i)
1786  continue
      do 1787 i=1,ndr
      bd(i)=adr(i)
1787  continue
      zc=za
      za=zb
      zb=zc
      goto 1901
781  continue
      call sl(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
785  continue
c    write(5,901) zc
901  format(1x,'знак суммы=',i2)
c    write(5,*) 'целая часть суммы'
c    write(5,445) (rezc(i),i=1,kcel)
c    write(5,*) 'дробная часть суммы'
c    write(5,445) (rezdr(i),i=1,kdro)
c    close(5,status='keep')
      n=ni
      nd=ndi
      m=mi
      md=mdi
      do 8721 i=1,n
      a(i)=ai(i)
8721  continue
      do 8722 i=1,nd
      ad(i)=adi(i)
8722  continue
      do 8723 i=1,m
      b(i)=bi(i)
8723  continue
      do 8724 i=1,md
      bd(i)=bdi(i)
8724  continue
      za=zais
      zb=zbis
      return
      end

```

## Набор wich.for

с-длинное вычитание

\$large

```

subroutine wich(n,nd,a,ad,m,md,b,bd,kcra,kdra,cra,dra,krazr)
integer*2 a(400),ad(400),b(400),bd(400),c(400),cd(400)
integer*2 cra(400),dra(400)
с integer a,ad,b,bd,c,cd,cra,dra
с open(5,file='dlinwich',status='Unknown')

с write(*,111)
111 format(1x,'введите длину целой части уменьшаемого')
с read(*,*) n
с write(*,112)
112 format(1x,'введите длину дробной части уменьшаемого')
с read(*,*) nd
с write(*,1111)
1111 format(1x,'введите длину целой части вычитаемого')
с read(*,*) m
с write(*,1112)
1112 format(1x,'введите длину дробной части вычитаемого')
с read(*,*) md

с write(*,334)
334 format(1x,'введите через пробел разряды целой части',
*' уменьшаемого')
с read(*,*) (a(i),i=1,n)
с write(*,335)
335 format(1x,'введите через пробел разряды дробной части',
*' уменьшаемого')

с read(*,*) (ad(i),i=1,nd)
с write(*,1334)
1334 format(1x,'введите через пробел разряды целой части',
*' вычитаемого')

с read(*,*) (b(i),i=1,m)
с write(*,1335)
1335 format(1x,'введите через пробел разряды дробной части',
*' вычитаемого')

с read(*,*) (bd(i),i=1,md)
с write(5,*) 'целая часть уменьшаемого'
с write(5,445) (a(i),i=1,n)
445 format(1x,79i1)
с write(5,*) 'дробная часть уменьшаемого'
с write(5,445) (ad(i),i=1,nd)
с write(5,*) 'целая часть вычитаемого'
с write(5,445) (b(i),i=1,m)
с write(5,*) 'дробная часть вычитаемого'
с write(5,445) (bd(i),i=1,md)
с if(krazr.le.100) goto 66791
с write(*,*) 'вычитание, количество разрядов>100!'
с pause
с stop
66791 continue
do 56881 i=1,400
cra(i)=0
56881 dra(i)=0
if(n.ge.m) goto 897

```

```

do 678 i=1,n
c(i)=a(i)
678 continue
do 679 i=1,nd
cd(i)=ad(i)
679 continue
do 680 i=1,m
a(i)=b(i)
680 continue
do 681 i=1,md
ad(i)=bd(i)
681 continue
do 682 i=1,n
b(i)=c(i)
682 continue
do 683 i=1,nd
bd(i)=cd(i)
683 continue
ir=n
n=m
m=ir
ir=nd
nd=md
md=ir
897 continue

```

с-формирование двух массивов для вычитания

```

max=nd
if(max.lt.md) max=md
if(md.eq.max) goto 78
in=md+1
do 79 i=in,max
bd(i)=0
79 continue
goto 81
78 continue
in=nd+1
do 82 i=in,max
ad(i)=0
82 continue
81 continue
it=0
in=n+1
ik=n+max
do 100 i=in,ik
it=it+1
a(i)=ad(it)
100 continue
it=0
in=m+1
ik=m+max
do 101 i=in,ik
it=it+1
b(i)=bd(it)
101 continue
i=0
nt=n+max
mt=m+max
10 continue
if(a(nt).ge.b(mt)) goto 1
np=nt-1
3 continue
if(np.eq.0) goto 201

```

```

        if(a(np).ne.0) goto 2
        np=np-1
        goto 3
2       continue
        a(np)=a(np)-1
7       continue
        np=np+1
        if(np.eq.nt) goto 30
        a(np)=9
        goto 7
30      continue
        a(nt)=a(nt)+10
        goto 10
1       continue
        i=i+1
        c(i)=a(nt)-b(mt)
        nt=nt-1
        mt=mt-1
        if(mt.ne.0) goto 10
        if(n.eq.m) goto 781
561     continue
        i=i+1
        c(i)=a(nt)
        nt=nt-1
        if(nt.ge.1) goto 561
781     continue
c       write(5,*) 'целая часть разности'
        in=i
        ik=max+1
c       write(5,445) (c(j),j=in,ik,-1)
c       write(5,*) 'дробная часть разности'
        in=max
c       write(5,445) (c(j),j=in,1,-1)
        goto 999
201     continue
        write(5,*) 'из меньшего вычитаем большее,здесь нельзя!'
        stop
999     continue
c       close(5,status='keep')
        kcra=i-max
        kdra=max
        it=0
        in=i
        ik=max+1
        do 2001 j=i,ik,-1
        it=it+1
        cra(it)=c(j)
2001    continue
        im=0
        do 3121 j=1,it
        if(cra(j).ne.0) goto 3122
        im=im+1
3121    continue
        goto 973
3122    continue
        il=0
        do 3123 jp=j,it
        il=il+1
        cra(il)=cra(jp)
3123    continue
973     continue
        kcra=kcra-im
        it=0

```

```
in=max
do 2002 j=max,1,-1
it=it+1
dra(it)=c(j)
2002 continue
if(kcra.ne.0) goto 971
kcra=1
cra(1)=0
971 continue
if(kdra.ne.0) goto 972
kdra=1
dra(1)=0
972 continue
if(kdra.gt.krazr) kdra=krazr
c if(kcra.le.krazr) goto 67291
c write(*,67292) kcra,krazr
67292 format(1x,'вычитание, количество разрядов в целой части=',i3,
*' больше заданного=',i3)
c pause
c stop
67291 continue

return
end
```

## Набор wiczn.for

```

$large
      subroutine wiczn(za,zb,zc,n,nd,a,ad,m,md,b,bd,
      *kcel,kdro,rezc,rezdr,krazr)
с-длинное вычитание с учетом знаков уменьшаемого и вычитаемого
      integer*2 a(400),b(400),c(400),ad(400),bd(400),cd(400)
      integer*2 rezc(400),rezdr(400),ar(400),adr(400)
      integer*2 ai(400),adi(400),bi(400),bdi(400)
      integer rez,os,p,za,zb,zc,zbi
      integer zais,zbis
      zais=za
      zbis=zb
      do 5671 i=1,n
      ai(i)=a(i)
5671  continue
      do 5672 i=1,nd
      adi(i)=ad(i)
5672  continue
      do 5673 i=1,m
      bi(i)=b(i)
5673  continue
      do 5674 i=1,md
      bdi(i)=bd(i)
5674  continue
      ni=n
      ndi=nd
      mi=m
      mdi=md

с      open(5,file='dlrazzn',status='Unknown')
с      write(*,2001)
2001  format(1x,'введите знак уменьшаемого,1-плюс,-1-минус')
с      read(*,*) za
с      write(*,2002)
2002  format(1x,'введите знак вычитаемого,1-плюс,-1-минус')
с      read(*,*) zb
      zbi=zb
      zb=zb*(-1)
с      write(*,111)
111  format(1x,'введите длину целой части уменьшаемого')
с      read(*,*) n
с      write(*,112)
112  format(1x,'введите длину дробной части уменьшаемого')
с      read(*,*) nd
с      write(*,1111)
1111  format(1x,'введите длину целой части вычитаемого')
с      read(*,*) m
с      write(*,1112)
1112  format(1x,'введите длину дробной части вычитаемого')
с      read(*,*) md

с      write(*,334)
334  format(1x,'введите через пробел разряды целой части',
      * ' уменьшаемого')
с      read(*,*) (a(i),i=1,n)
с      write(*,335)
335  format(1x,'введите через пробел разряды дробной части',
      * ' уменьшаемого')

с      read(*,*) (ad(i),i=1,nd)
с      write(*,1334)

```

```

1334  format(1x,'введите через пробел разряды целой части',
      * ' вычитаемого ')

с      read(*,*) (b(i),i=1,m)
с      write(*,1335)
1335  format(1x,'введите через пробел разряды дробной части',
      * ' вычитаемого ')

с      read(*,*) (bd(i),i=1,md)
с      write(5,*) 'целая часть уменьшаемого'
с      write(5,445) (a(i),i=1,n)
445   format(1x,70i1)
с      write(5,*) 'дробная часть уменьшаемого'
с      write(5,445) (ad(i),i=1,nd)
с      write(5,*) 'целая часть вычитаемого'
с      write(5,445) (b(i),i=1,m)
с      write(5,*) 'дробная часть вычитаемого'
с      write(5,445) (bd(i),i=1,md)
      ip=za*zb
      if(ip.gt.0) goto 781
      if(za.eq.1) goto 782
1901  if(m.gt.n) goto 783
      if(m.eq.n) goto 784
      call wich(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
      goto 785
784   continue
      do 786 i=1,m
      if(a(i).gt.b(i)) goto 787
      if(a(i).lt.b(i)) goto 788
786   continue
      nmin=nd
      if(nmin.gt.md) nmin=md
      do 791 i=1,nmin
      if(ad(i).gt.bd(i)) goto 792
      if(ad(i).lt.bd(i)) goto 793
791   continue
      if(nd.le.md) goto 795
      call wich(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
      goto 785
795   continue
      call wich(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
      zc=zb
      goto 785
793   continue
      call wich(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
      zc=zb
      goto 785
792   continue
      call wich(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
      goto 785
788   continue
      call wich(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
      zc=zb
      goto 785
787   continue
      call wich(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
      goto 785
783   continue

```

```

      call wich(m,md,b,bd,n,nd,a,ad,kcel,kdro,rezc,rezdr,krazr)
      zc=zb
      goto 785
782  continue
      do 1782 i=1,n
      ar(i)=a(i)
1782  continue
      do 1783 i=1,nd
      adr(i)=ad(i)
1783  continue
      nr=n
      ndr=nd
      do 1784 i=1,m
      a(i)=b(i)
1784  continue
      do 1785 i=1,md
      ad(i)=bd(i)
1785  continue
      n=m
      nd=md
      m=nr
      md=ndr
      do 1786 i=1,nr
      b(i)=ar(i)
1786  continue
      do 1787 i=1,ndr
      bd(i)=adr(i)
1787  continue
      zc=za
      za=zb
      zb=zc
      goto 1901
781  continue
      call sl(n,nd,a,ad,m,md,b,bd,kcel,kdro,rezc,rezdr,krazr)
      zc=za
785  continue
c    write(5,901) zc
901  format(1x,'знак разности=',i2)
      zb=zbi

      n=ni
      nd=ndi
      m=mi
      md=mdi
      do 8721 i=1,n
      a(i)=ai(i)
8721  continue
      do 8722 i=1,nd
      ad(i)=adi(i)
8722  continue
      do 8723 i=1,m
      b(i)=bi(i)
8723  continue
      do 8724 i=1,md
      bd(i)=bdi(i)
8724  continue
      za=zais
      zb=zbis
      return
      end

```

**Набор ymnzn.for**

```

$large
  subroutine ymnzn(za,zb,zc,n,nd,a,ad,m,md,b,bd,
    *kcel,kdro,ctot,cdtot,krazr)
с-длинное умножение с учетом знаков сомножителей
  integer*2 a(400),b(400),c(400),ad(400),bd(400),cd(400)
  integer*2 cdtot(400),ctot(400),csl(400),csld(400)
  integer*2 rezc(400),rezdr(400)
  integer*2 ai(400),adi(400),bi(400),bdi(400)
  integer rez,os,p,pr
  integer za,zb,zc,zais,zbis
  zais=za
  zbis=zb
  do 5671 i=1,n
    ai(i)=a(i)
5671 continue
  do 5672 i=1,nd
    adi(i)=ad(i)
5672 continue
  do 5673 i=1,m
    bi(i)=b(i)
5673 continue
  do 5674 i=1,md
    bdi(i)=bd(i)
5674 continue
  ni=n
  ndi=nd
  mi=m
  mdi=md

  zc=za*zb
с open(5,file='dlinymno',status='Unknown')
с write(*,111)
111 format(1x,'введите длину целой части первого множителя')
с read(*,*) n
с write(*,112)
112 format(1x,'введите длину дробной части первого множителя')
с read(*,*) nd
с write(*,1111)
1111 format(1x,'введите длину целой части второго множителя')
с read(*,*) m
с write(*,1112)
1112 format(1x,'введите длину дробной части второго множителя')
с read(*,*) md

с write(*,334)
334 format(1x,'введите через пробел разряды целой части',
*' первого множителя')
с read(*,*) (a(i),i=1,n)
с write(*,335)
335 format(1x,'введите через пробел разряды дробной части',
*' первого множителя')

с read(*,*) (ad(i),i=1,nd)
с write(*,1334)
1334 format(1x,'введите через пробел разряды целой части',
*' второго множителя')

с read(*,*) (b(i),i=1,m)
с write(*,1335)
1335 format(1x,'введите через пробел разряды дробной части',

```

```

* ' второго множителя' )

с read(*,*) (bd(i),i=1,md)
с write(5,*) 'целая часть первого множителя'
с write(5,445) (a(i),i=1,n)
445 format(1x,70i1)
с write(5,*) 'дробная часть первого множителя'
с write(5,445) (ad(i),i=1,nd)
с write(5,*) 'целая часть второго множителя'
с write(5,445) (b(i),i=1,m)
с write(5,*) 'дробная часть второго множителя'
с write(5,445) (bd(i),i=1,md)

call ymn(n,nd,a,ad,m,md,b,bd,kcel,kdro,ctot,cdtot,krazr)

с write(5,*) 'целая часть произведения'
с write(5,445) (ctot(i),i=1,kcel)
с write(5,*) 'дробная часть произведения'
с write(5,445) (cdtot(i),i=1,kdro)
с close(5,status='keep')
n=ni
nd=ndi
m=mi
md=mdi
do 8721 i=1,n
a(i)=ai(i)
8721 continue
do 8722 i=1,nd
ad(i)=adi(i)
8722 continue
do 8723 i=1,m
b(i)=bi(i)
8723 continue
do 8724 i=1,md
bd(i)=bdi(i)
8724 continue
za=zais
zb=zbis

return
end

```

## сумн

с-длинное умножение  
\$large

```

subroutine ymn(n,nd,a,ad,m,md,b,bd,kcel,kdro,ctot,cdtot,krazr)
integer*2 a(400),b(400),c(400),ad(400),bd(400),cd(400)
integer*2 cdtot(400),ctot(400),csl(400),csld(400)
integer*2 rezc(400),rezdr(400)
integer rez,os,p,pr
с integer rezc,rezdr
с open(5,file='ymno',status='Unknown')
с write(*,111)
111 format(1x,'введите длину целой части первого множителя')
с read(*,*) n
с write(*,112)
112 format(1x,'введите длину дробной части первого множителя')
с read(*,*) nd
с write(*,1111)
1111 format(1x,'введите длину целой части второго множителя')

```

```

с      read(*,*) m
с      write(*,1112)
1112   format(1x,'введите длину дробной части второго множителя')
с      read(*,*) md

с      write(*,334)
334   format(1x,'введите через пробел разряды целой части',
*' первого множителя')
с      read(*,*) (a(i),i=1,n)
с      write(*,335)
335   format(1x,'введите через пробел разряды дробной части',
*' первого множителя')

с      read(*,*) (ad(i),i=1,nd)
с      write(*,1334)
1334   format(1x,'введите через пробел разряды целой части',
*' второго множителя')

с      read(*,*) (b(i),i=1,m)
с      write(*,1335)
1335   format(1x,'введите через пробел разряды дробной части',
*' второго множителя')

с      read(*,*) (bd(i),i=1,md)
с      write(5,*) 'целая часть первого множителя'
с      write(5,445) (a(i),i=1,n)
445   format(1x,70i1)
с      write(5,*) 'дробная часть первого множителя'
с      write(5,445) (ad(i),i=1,nd)
с      write(5,*) 'целая часть второго множителя'
с      write(5,445) (b(i),i=1,m)
с      write(5,*) 'дробная часть второго множителя'
с      write(5,445) (bd(i),i=1,md)
с      if(krazr.le.100) goto 66791
с      write(*,*) 'умножение, количество разрядов>100!'
с      pause
с      stop
66791 continue
      do 66987 i=1,400
          ctot(i)=0
66987 cdtot(i)=0
с-формирование двух массивов для умножения
      it=0
      in=n+1
      ik=n+nd
      do 100 i=in,ik
          it=it+1
          a(i)=ad(it)
100   continue
      it=0
      in=m+1
      ik=m+md
      do 101 i=in,ik
          it=it+1
          b(i)=bd(it)
101   continue
      ntot=n+m
      kdtot=nd+md
      do 651 i=1,ntot
          ctot(i)=0
651   continue

```

```

        do 652 i=1,kdtot
        cdtot(i)=0
652   continue

с-умножение
        i=0
        im=0
        p=0
        nt=n+nd
        mt=m+md
801   continue
        p=0
        do 941 jp=1,400
        c(jp)=0
941   continue
        i=0
        nt=n+nd
        im=im+1
701   continue
        pr=a(nt)*b(mt)+p
        i1=pr/10
        os=pr-i1*10
        i=i+1
        c(i)=os
        p=i1
        nt=nt-1
        if(nt.ge.1) goto 701
        if(p.eq.0) goto 1701
        i=i+1
        c(i)=p
        p=0
1701  continue

c      write(5,942) im,i,(c(jp),jp=1,i)
942   format(1x,'im=',i3,'i dlj c=',i3/1x,79i1)
        kraz=md-im+1
        kcd=nd+kraz
        icel=i-kcd
        if(icel.lt.i) goto 1921
        iab=abs(kcd)
        it=0
        do 1922 jp=i,1,-1
        it=it+1
        csl(it)=c(jp)
1922  continue
        if(iab.eq.0) goto 1941
        do 1923 jp=iab,1,-1
        it=it+1
        csl(it)=0
1923  continue
1941  continue
        idr=1
        csld(1)=0
        goto 8722
1921  continue

        it=0
        in=i
        ik=kcd+1
        if(in.eq.ik) goto 961
        if(in.lt.ik) goto 962
        do 702 j=in,ik,-1
        it=it+1

```

```

csl(it)=c(j)
702 continue
icel=i-kcd
goto 963
961 continue
icel=1
csl(1)=c(in)
goto 963
962 continue
icel=1
csl(1)=0
963 continue
in=kcd
it=0
do 703 j=in,1,-1
it=it+1
csld(it)=c(j)
703 continue
idr=kcd
if(idr.gt.0) goto 8722
idr=1
csld(1)=0
8722 continue
c write(5,*) 'целая часть первого слагаемого'
c write(5,445) (csl(j),j=1,icel)
c write(5,*) 'дробная часть первого слагаемого'
c write(5,445) (csld(j),j=1,idr)
c write(5,*) 'целая часть второго слагаемого'
c write(5,445) (ctot(j),j=1,ntot)
c write(5,*) 'дробная часть второго слагаемого'
c write(5,445) (cdtot(j),j=1,kdtot)

call sl(icel,idr,csl,csld,ntot,kdtot,ctot,cdtot,kcel,kdro,
*rezc,rezdr,krazr)
if(kcel.ne.0) goto 8723
kcel=1
rezc(1)=0
8723 continue
if(kdro.ne.0) goto 8724
kdro=1
rezdr(1)=0
8724 continue
c write(5,*) 'целая часть суммы'
c write(5,445) (rezc(j),j=1,kcel)
c write(5,*) 'дробная часть суммы'
c write(5,445) (rezdr(j),j=1,kdro)

do 704 i=1,kcel
ctot(i)=rezc(i)
704 continue
do 705 i=1,kdro
cdtot(i)=rezdr(i)
705 continue
ntot=kcel
kdtot=kdro
mt=mt-1
if(mt.ge.1) goto 801
c write(5,*) 'целая часть произведения'
c write(5,445) (ctot(i),i=1,kcel)
c write(5,*) 'дробная часть произведения'
c write(5,445) (cdtot(i),i=1,kdro)
c close(5,status='keep')
it=0

```

```
do 7766 j=1,kcel
  if(ctot(j).ne.0) goto 7767
  it=it+1
7766 continue
7767 continue
  il=0
  do 7768 jp=j,kcel
  il=il+1
  ctot(il)=ctot(jp)
7768 continue
  kcel=kcel-it
  if(kcel.ne.0) goto 3999
  kcel=1
  ctot(1)=0
3999 continue
  if(kdro.ne.0) goto 4000
  kdro=1
  cdtot(1)=0
4000 continue

  if(kdro.gt.krazr) kdro=krazr
c   if(kcel.le.krazr) goto 67291
c   write(*,67292) kcel,krazr
67292 format(1x,'умножение, количество разрядов в целой части=',i3,
  *' больше заданного=',i3)
c   pause
c   stop
67291 continue

return
end
```

## Набор delzn.for

с-длинное деление с учетом знаков

\$large

```

subroutine delzn(za,zb,zc,n,nd,a,ad,m,md,b,bd,
*kcdel,kddel,cdel,ddel,krazr)
integer*2 a(400),ad(400),b(400),bd(400),c(400),cd(400)
integer*2 ar(400),adr(400),br(400),bdr(400)
integer*2 cdel(400),ddel(400),ctot(400),cdtot(400)
integer*2 rezc(400),rezdr(400),ce(400),ced(400)
c integer a,ad,b,bd,c,cd,cdel,ddel,ctot,cdtot,rezc,rezdr
integer za,zb,zc,zais,zbis
zais=za
zbis=zb
do 5671 i=1,n
ar(i)=a(i)
5671 continue
do 5672 i=1,nd
adr(i)=ad(i)
5672 continue
do 5673 i=1,m
br(i)=b(i)
5673 continue
do 5674 i=1,md
bdr(i)=bd(i)
5674 continue
ni=n
ndi=nd
mi=m
mdi=md
zc=za*zb
c open(5,file='dlindel',status='Unknown')
c write(*,111)
111 format(1x,'введите длину целой части делимого')
c read(*,*) n
c write(*,112)
112 format(1x,'введите длину дробной части делимого')
c read(*,*) nd
c write(*,1111)
1111 format(1x,'введите длину целой части делителя')
c read(*,*) m
c write(*,1112)
1112 format(1x,'введите длину дробной части делителя')
c read(*,*) md

c write(*,334)
334 format(1x,'введите через пробел разряды целой части',
*' делимого')
c read(*,*) (a(i),i=1,n)
c write(*,335)
335 format(1x,'введите через пробел разряды дробной части',
*' делимого')

c read(*,*) (ad(i),i=1,nd)
c write(*,1334)
1334 format(1x,'введите через пробел разряды целой части',
*' делителя')

c read(*,*) (b(i),i=1,m)
c write(*,1335)
1335 format(1x,'введите через пробел разряды дробной части',
*' делителя')

```

```

c      read(*,*) (bd(i),i=1,md)
c      write(5,*) 'целая часть делимого'
c      write(5,445) (a(i),i=1,n)
445   format(1x,79i1)
c      write(5,*) 'дробная часть делимого'
c      write(5,445) (ad(i),i=1,nd)
c      write(5,*) 'целая часть делителя'

c      write(5,445) (b(i),i=1,m)
c      write(5,*) 'дробная часть делителя'
c      write(5,445) (bd(i),i=1,md)

      call deld(n,nd,a,ad,m,md,b,bd,kcdel,kddel,cdel,ddel,krazr)

c      write(5,*) 'целая часть частного'
c      write(5,445) (cdel(j),j=1,kcdel)
c      write(5,*) 'дробная часть частного'
c      write(5,445) (ddel(j),j=1,kddel)
c      close(5,status='keep')
      n=ni
      nd=ndi
      m=mi
      md=mdi
      do 8721 i=1,n
      a(i)=ar(i)
8721  continue
      do 8722 i=1,nd
      ad(i)=adr(i)
8722  continue
      do 8723 i=1,m
      b(i)=br(i)
8723  continue

      do 8724 i=1,md
      bd(i)=bdr(i)
8724  continue
      za=zais
      zb=zbis
      return
      end

```

## cdeld

\$large

```

      subroutine deld(n,nd,a,ad,m,md,b,bd,kcdel,kddel,cdel,ddel,krazr)
c-длинное деление
      integer*2 a(400),ad(400),b(400),bd(400),c(400),cd(400)
      integer*2 cdel(400),ddel(400),ctot(400),cdtot(400)
      integer*2 rezc(400),rezdr(400),ce(400),ced(400)
c      integer a,ad,b,bd,c,cd,cdel,ddel,ctot,cdtot,rezc,rezdr
c      integer ce,ced
c      if(krazr.le.100) goto 66791
c      write(*,*) 'деление, количество разрядов>100!'
c      pause
c      stop
66791 continue
      do 66789 i=1,400
      cdel(i)=0
66789 ddel(i)=0

```

c-формирование двух массивов для деления

```

max=nd
if(max.lt.md) max=md
if(md.eq.max) goto 78
in=md+1
do 79 i=in,max
bd(i)=0
79 continue
goto 81
78 continue
in=nd+1
do 82 i=in,max
ad(i)=0
82 continue
81 continue
it=0
in=n+1
ik=n+max
do 100 i=in,ik
it=it+1
a(i)=ad(it)
100 continue
il=0
do 5671 j=1,ik
if(a(j).ne.0) goto 5672
il=il+1
5671 continue
kcdel=0
kddel=0
goto 1231
stop
5672 continue
it=0
do 5673 jp=j,ik
it=it+1
a(it)=a(jp)
5673 continue
it=0
in=m+1
ik=m+max
do 101 i=in,ik
it=it+1
b(i)=bd(it)
101 continue
ill=0
do 15671 j=1,ik
if(b(j).ne.0) goto 15672
ill=ill+1
15671 continue
write(6,*) 'на ноль делить нельзя'
stop
15672 continue
it=0
do 15673 jp=j,ik
it=it+1
b(it)=b(jp)
15673 continue

c      if(n.lt.m) goto 675
c      if(n.eq.m) goto 676
do 2677 i=1,400
ctot(i)=0
cdtot(i)=0
2677 continue

```

```

kctot=1
kdto=1
ce(1)=1
kedi=1
ced(1)=0
kedid=1
kad=1
kbd=1
ad(1)=0
bd(1)=0
nat=n+max-il
mat=m+max-ill
isd=0
kcel=1
kdro=1
rezc(1)=0
rezdr(1)=0
2010  continue
c  write(5,4501) nat,mat
4501  format(1x,'nat=',i3,' mat=',i3)
      if(nat.lt.mat) goto 675
      if(nat.eq.mat) goto 676
      goto 677
675   continue
      if(isd.ne.0) goto 5621
      do 5701 j=1,kcel
        cdel(j)=rezc(j)
5701  continue
      do 5702 j=1,kdro
        ddel(j)=rezdr(j)
5702  continue
      rezc(1)=0
      kcdel=kcel
      kddel=kdro
      if(nat.eq.0) goto 1231
5621  continue
      if(isd.eq.0) goto 8721
c  write(5,3451) isd,rezc(1)
3451  format(1x,'isd=',i3,' rezc(1)=' ,i3)
      ddel(isd)=rezc(1)
      rezc(1)=0
      kddel=isd
8721  continue
c  nat=nat+1
c  isd=isd+1
c  kddel=isd
      if(nat.eq.0) goto 1231
      do 9781 j=1,nat
c  write(5,9788) nat,isd,j,a(j)
9788  format(1x,'nat=',i3,' isd=',i3,' j=',i3,' a(j)=' ,i3)
      if(a(j).ne.0) goto 9782
9781  continue
      goto 1231
9782  continue
      nat=nat+1
      isd=isd+1
      if(isd.gt.krazr) goto 1231
      a(nat)=0
      do 1677 i=1,400
        ctot(i)=0
        cdtot(i)=0
1677  continue
      kctot=1

```

```

        kdto=1
        goto 2010
676   continue
        do 4562 j=1,nat
        if(a(j).gt.b(j)) goto 677
        if(a(j).lt.b(j)) goto 675
4562  continue
677   continue
        call wich(nat,kad,a,ad,mat,kbd,b,bd,kc,kcd,c,cd,krazr)
        call sl(kctot,kdto,ctot,cdtot,kedi,kedid,ce,ced,
        *kcel,kdro,rezc,rezdr,krazr)
        kctot=kcel
        kdto=kdro
        do 1121 i=1,kcel
        ctot(i)=rezc(i)
1121  continue
        do 1122 i=1,kdro
        cdtot(i)=rezdr(i)
1122  continue
        nat=kc
        kad=kcd
        if(nat.eq.0) a(1)=0
        do 3001 j=1,nat
c      if(nat.ne.0) goto 3112
c      write(5,3113) nat,j,a(j),c(j)
3113  format(1x,'nat=',i3,' j=',i3,' a(j)=',i3,' c(j)=',i3)
3112  continue
        a(j)=c(j)
3001  continue
        do 3002 j=1,kad
        ad(j)=cd(j)
3002  continue
        goto 2010
1231  continue
        if(kcdel.ne.0) goto 971
        kcdel=1
        cdel(1)=0
971   continue
        if(kddel.ne.0) goto 972
        kddel=1
        ddel(1)=0
972   continue
        if(kddel.gt.krazr) kddel=krazr
c      if(kcdel.le.krazr) goto 67291
c      write(*,67292) kcdel,krazr
67292 format(1x,'деление, количество разрядов в целой части=',i3,
        *' больше заданного=',i3)
c      pause
c      stop
67291 continue

c      write(5,*) 'целая часть частного'
c      write(5,445) (cdel(j),j=1,kcdel)
c      write(5,*) 'дробная часть частного'
c      write(5,445) (ddel(j),j=1,kddel)
c      close(5,status='keep')
        return
        end

```

**Файл name(названия параметров-столбиков матрицы исследования)**

5  
 (p1)  
 (p2)  
 (p3)  
 (p4)  
 (p5)

**Файл sled(матрица исследования)**

(проба длинная мнк)

4 5  
 1 0 1 7 4  
 3 1 3 1 2  
 7 2 9 3 2  
 13 3 1 2 7

**Файл dlmnk2(полином,вклады параметров-аргументов)**

(проба длинная мнк)

	1	1	1
целая часть			
1			
дробная часть			
0			
	1	1	1
целая часть			
0			
дробная часть			
0			
	1	1	1
целая часть			
1			
дробная часть			
0			
	1	1	1
целая часть			
7			
дробная часть			
0			
	1	1	1
целая часть			
4			
дробная часть			
0			
	1	1	1
целая часть			
3			
дробная часть			
0			
	1	1	1
целая часть			
1			
дробная часть			
0			
	1	1	1
целая часть			
3			
дробная часть			
0			
	1	1	1

целая часть			
1			
дробная часть			
0			
1	1	1	1
целая часть			
2			
дробная часть			
0			
1	1	1	1
целая часть			
7			
дробная часть			
0			
1	1	1	1
целая часть			
2			
дробная часть			
0			
1	1	1	1
целая часть			
9			
дробная часть			
0			
1	1	1	1
целая часть			
3			
дробная часть			
0			
1	1	1	1
целая часть			
2			
дробная часть			
0			
1	2	1	1
целая часть			
13			
дробная часть			
0			
1	1	1	1
целая часть			
3			
дробная часть			
0			
1	1	1	1
целая часть			
1			
дробная часть			
0			
1	1	1	1
целая часть			
2			
дробная часть			
0			
1	1	1	1
целая часть			
7			
дробная часть			
0			

степень полинома= 1

зависимый параметр- 1  
(p1)

номера аргументов  
4 5

оптимальная степень полинома= 1

Модель для зависимого параметра

аргумент- 4  
(p4)  
номер аргумента- 4 степень= 1  
коэффициент  
знак=-1  
целая часть коэффициента

1  
дробная часть коэффициента  
10693641618497109810

аргумент- 5  
(p5)  
номер аргумента- 5 степень= 1  
коэффициент  
знак= 1  
целая часть коэффициента

1  
дробная часть коэффициента  
57514450867052023133

свободный член модели

знак свободного члена= 1  
целая часть свободного члена  
3  
дробная часть свободного члена  
69075144508670520128  
средняя абсолютная ошибка  
целая часть ошибки

1  
дробная часть ошибки  
98843930635838150292

номер аргумента= 4  
(p4)  
целая часть вклада

0  
дробная часть вклада  
37851604074806142612

номер аргумента= 5  
(p5)  
целая часть вклада

0  
дробная часть вклада  
62148395925193857387

## Файл proowfdl2(проверка решения системы нормальных уравнений)

```

проверка решения матрица системы
элемент с( 1, 1)
знак= 1
длина целой= 2 длина дробной= 4
целая часть=63
дробная часть=0000
элемент с( 1, 2)
знак= 1
длина целой= 2 длина дробной= 4
целая часть=50
дробная часть=0000
элемент с( 1, 3)
знак= 1
длина целой= 2 длина дробной= 2
целая часть=13
дробная часть=00
элемент с( 2, 1)
знак= 1
длина целой= 2 длина дробной= 4
целая часть=50
дробная часть=0000
элемент с( 2, 2)
знак= 1
длина целой= 2 длина дробной= 4
целая часть=73
дробная часть=0000
элемент с( 2, 3)
знак= 1
длина целой= 2 длина дробной= 2
целая часть=15
дробная часть=00
элемент с( 3, 1)
знак= 1
длина целой= 2 длина дробной= 2
целая часть=13
дробная часть=00
элемент с( 3, 2)
знак= 1
длина целой= 2 длина дробной= 2
целая часть=15
дробная часть=00
элемент с( 3, 3)
знак= 1
длина целой= 1 длина дробной= 1
целая часть=4
дробная часть=0
свободные члены
i= 1
знак= 1
длина целой= 2 длина дробной= 3
целая часть=57
дробная часть=000
i= 2
знак= 1
длина целой= 3 длина дробной= 3
целая часть=115
дробная часть=000
i= 3
знак= 1
длина целой= 2 длина дробной= 1
целая часть=24

```

дробная часть=0  
 неизвестные системы  
 i= 1  
 знак= -1  
 длина целой= 1 длина дробной= 20  
 целая часть=1  
 дробная часть=10693641618497109810  
 i= 2  
 знак= 1  
 длина целой= 1 длина дробной= 20  
 целая часть=1  
 дробная часть=57514450867052023133  
 i= 3  
 знак= 1  
 длина целой= 1 длина дробной= 20  
 целая часть=3  
 дробная часть=69075144508670520128  
 уравнение- 1  
 правая часть, исход  
 знак= 1  
 длина целой= 2 длина дробной= 3  
 целая часть=57  
 дробная часть=000  
 правая часть, счет  
 знак= 1  
 длина целой= 2 длина дробной= 20  
 целая часть=57  
 дробная часть=00000000000000000284  
 разница  
 знак= 1  
 длина целой= 1 длина дробной= 20  
 целая часть=0  
 дробная часть=00000000000000000284  
 уравнение- 2  
 правая часть, исход  
 знак= 1  
 длина целой= 3 длина дробной= 3  
 целая часть=115  
 дробная часть=000  
 правая часть, счет  
 знак= 1  
 длина целой= 3 длина дробной= 20  
 целая часть=115  
 дробная часть=00000000000000000129  
 разница  
 знак= 1  
 длина целой= 1 длина дробной= 20  
 целая часть=0  
 дробная часть=00000000000000000129  
 уравнение- 3  
 правая часть, исход  
 знак= 1  
 длина целой= 2 длина дробной= 1  
 целая часть=24  
 дробная часть=0  
 правая часть, счет  
 знак= 1  
 длина целой= 2 длина дробной= 20  
 целая часть=23  
 дробная часть=9999999999999999977  
 разница  
 знак= -1  
 длина целой= 1 длина дробной= 20

целая часть=0  
дробная часть=0000000000000000000023

## ПРИЛОЖЕНИЕ-2(наборы с исходными текстами программ на C++, исходные и выходные данные для программы на C++)

### Набор dlmnkbc.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlmnkbc.h"
#include<math.h>
#include<alloc.h>
LRESULT CALLBACK WindowFunc (HWND,UINT,WPARAM,LPARAM);
BOOL CALLBACK DialogFunc (HWND,UINT,WPARAM,LPARAM);

char szWinName[]="|юх юъэю";/*шь ьырөөр юъэр*/
HINSTANCE hInst;
int i61=0,i62=0,i63=0,i64=0,i65=0,i66=0,i67=0,i68=0;
int ikr=10;
int irasp=0,igraf=0,idin=0,ipampe,knaz,ktob;
char **anaz;
// int xstr=0,ystr=0;/*Схъе
//шх ьююЕфшэрСв әСЕюш */
int maxхе,махе;/*ЕрчьхЕв ьЕрэр */
// char strw[255];/* +еІхЕ әСЕюш твтюфр */
FILE *fp6,*fp4,*fp5,*fp7,*fp9;
int WINAPI WinMain(HINSTANCE hThisInst,
                   HINSTANCE hPrevInst,
                   LPSTR lpszArgs,
                   int nWinMode)
{
    HWND hwnd;
    MSG msg;
    WNDCLASS wcl;
    HACCEL hAccel;
/*юяЕхфхьшСН ьырөө юъэра*/
wcl.hInstance=hThisInst;/*фхъЕшяСюЕ фрээюую яЕшьюцхэш */
wcl.lpszClassName=szWinName;/* шь ьырөөр юъэр*/
wcl.lpfnWndProc=WindowFunc;/*іеэьҮш юъэр*/
wcl.style=0; /*әСшыН яю ьююҮрэш|*/
wcl.hIcon=LoadIcon(NULL,IDI_APPLICATION);/*әСрәфрЕСәр шьюәэр*/
wcl.hCursor=LoadCursor(NULL,IDC_ARROW);/*әСрәфрЕСәвц ьеЕёюЕ*/
/*чфхәН ьрчвтрхСә шь ьхә|*/
wcl.lpszMenuName="MYMENU";/*L|- ьхә|*/
wcl.cbClsExtra=0;/*схч фюяюыәшСхыНәюц*/
wcl.cbWndExtra=0;/*шәІюЕьрҮшш*/
/* юяЕхфхьшСН чряюыәхэшх юъэр схыв ҮтхСюь*/
wcl.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
/* чрЕхушәСЕшЕютрСН ьырөө юъэр*/
if(!RegisterClass(&wcl))
    return 0;
/* әючфрСН юъэю*/
hwnd=CreateWindow(szWinName,/* шь ьырөөр юъэр*/
                 "dlmnkbc",/* чруююютюь*/
                 WS_OVERLAPPEDWINDOW,/* әСшыН юъэр*/
                 CW_USEDEFAULT,/* X-ьююЕфшэрСр юяЕхфхы хСә Windows*/
                 CW_USEDEFAULT,/* Y-ьююЕфшэрСр юяЕхфхы хСә Windows*/
                 CW_USEDEFAULT,/* °шЕшәр юяЕхфхы хСә Windows*/
                 CW_USEDEFAULT,/* твёюСр юяЕхфхы хСә Windows*/
                 HWND_DESKTOP,/* эхС ЕюфшСхыНәююю юъэр */
                 NULL,/* эхС ьхә|*/
                 hThisInst,/*фхъЕшяСюЕ яЕшьюцхэш */

```

```

        NULL/* счх фюяюыэшсхын#эви рёуеьжэсют*/
    );
    /* чруёечшсн# рьёыхьёрсюёв */
    hAccel=LoadAccelerators(hThisInst,"MYMENU");
    /* яюьрчрсн# юъэю ш яхёхёшёютрсн# ёюфхёцшьюк */
    ShowWindow(hwnd,nWinMode);
    UpdateWindow(hwnd);
    /* чряёёсшсн# ўшгы юсёрсюёъш ёююс
хэщц*/
    while(GetMessage(&msg,NULL,0,0))
    {
        if(!TranslateAccelerator(hwnd,hAccel,&msg));
        {
            TranslateMessage(&msg);/* ёрчёх °шсн# шёяюын#чютрэшх ьыртшрсёёв*/
            DispatchMessage(&msg);/* тхёэесн# еяёртыхэшх Windows*/
        }
    }
    return msg.wParam;
}
/* їезьўш юъэр */
LRESULT CALLBACK WindowFunc(HWND hwnd,UINT message,
                               WPARAM wParam,
                               LPARAM lParam)
{
extern void dlmnkbc();
    HDC hdc;
    TEXTMETRIC tmet;
    SIZE size;
    int response;
    switch(message)
    {
        case WM_CREATE:
            /* |+TL+L+| |L||+|- |||L=L */
            maxxe=GetSystemMetrics(SM_CXSCREEN);
            maxye=GetSystemMetrics(SM_CYSCREEN);
            break;
        case WM_COMMAND:
            switch(LOWORD(wParam))
            {
                case ID_RUN:
                    hdc=GetDC(hwnd);/*яюыёўшсн# ьюэсхъёс еёсёюощёстр */
                    /* еёсрэютшсн# ўхёэвц ўтхёсхъёср */
                    SetTextColor(hdc,RGB(0,0,0));
                    /* LTTL=+TLT- -||+--+TL++г гT+T L+=L */
                    SetBkColor(hdc,RGB(0,255,255));
                    /*|+TL+LT- |+T|L|L T+|TTL */
                    GetTextMetrics(hdc,&tmet);
                    // break;
                    // case ID_RUN:
                    MessageBox(hwnd,"press OK,wait,CALCULATE!", "",MB_OK);
                    dlmnkbc();
                    ReleaseDC(hwnd,hdc);/*юётюсюфшсн# ьюэсхъёс еёсёюощёстр */
                    MessageBox(hwnd,"END,PRESS OK", "",MB_OK);
                    break;
                case ID_OBRAZ:
                    DialogBox(hInst,"MYDB",hwnd,(DLGPROC)DialogFunc);
                    break;
                case ID_PORIW:
                    MessageBox(hwnd,"|юёвт ецх ёрёёьюсёёхэ","Т юсёрчўх",MB_OK);
                    break;
            }
            break;
    }
}

```

```

    case WM_DESTROY: /*чрҗхЕ°хэжх яЁюуЁрьёв*/
        PostQuitMessage(0);
        break;
    default:
        /*тёх ёююс
        хэш ,эх юЁрсрЕвтрхёвх т фрээюц Ёеэёушш,
        эряёрты |Её эр юЁрсюёёе яю ёьююёрэш|*/
        return DefWindowProc(hwnd,message,wParam,lParam);
    }
    return 0;

}

/* яЁюёср Ёеэёуш фшрыюур */
BOOL CALLBACK DialogFunc(HWND hwnd,

                                                                    UINT message,
                                                                    WPARAM wParam,
                                                                    LPARAM lParam)

{
// double i61,i62,i63,irasp,igraf,idin;
    switch(message)
    {
        case WM_COMMAND:
            switch(LOWORD(wParam))
            {

                case IDCANCEL:
                    EndDialog(hwnd,0);
                    return 1;
                case IDOK:
                    i61=SendDlgItemMessage(hwnd,P_1,BM_GETCHECK,0,0);
                    irasp=i61;
//                    i62=SendDlgItemMessage(hwnd,P_2,BM_GETCHECK,0,0);
//                    idin=i62;
                    i63=SendDlgItemMessage(hwnd,P_3,BM_GETCHECK,0,0);
                    igraf=i63;
//                    fprintf(fp,"dial irasp=%lf idin=%lf igraf=%lf\n",irasp,idin,igraf);
                    return 1;
            }
        }
    }
    return 0;
}

```

## Набор dlmnkbc.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlmnkbc.h"
#include<math.h>
#include<alloc.h>

//      FILE *fp4,*fp6,*fp5,*fp7,*fp9;

void dlmnkbc()
{
extern FILE *fp4,*fp6,*fp5,*fp7,*fp9;

extern void mnk(int n,int karg,int izn,
    int kcn,int kdn,int *icn,int *idn,int *izx,int *kcx,int *kdx,
    int *icx,int *idx,
    int *izy,int *kcy,int *kdy,int *icy,int *idy,int k,
    int *izak,int *kcak,int *kdak,
    int *icak,int *idak,int *pigau,int krazr,int k50);

    extern void ymnzn(int izw,int izr,int *pizrez,
        int kcw,int kdw,int *icw,int *idw,
        int kcr,int kdr,int *icr,int *idr,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void wiczn(int izr,int izrl,int *pizrez,
        int kcr,int kdr,int *icr,int *idr,

        int kcl,int kdl,int *icl,int *idl,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void slzn(int izr,int izrez,int *pizrl,
        int kcr,int kdr,int *icr,int *idr,
        int kcrez,int kdrez,int *icrez,int *idrez,
        int *pkcl,int *pkdl,int *icl,int *idl,int krazr);

    extern void delzn(int izr,int izn,int *pizrez,
        int kcr,int kdr,int *icr,int *idr,
        int kcn,int kdn,int *icn,int *idn,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

//с-метод наименьших квадратов в длинной арифметике
    char anaz[101][61];int mnarg[21];char zgl[81];
/*
    integer*2 izrab(50,5),kcrab(50,20),kdrab(50,20),
    *icrab(50,20,20),idrab(50,20,20),izy(50),kcy(50),kdy(50),

    *icy(50,20),idy(50,20),izx(50,10),kcx(50,10),kdx(50,10),
    *icx(50,10,20),idx(50,10,20),izakm(3,31),izak(31),kcakm(3,31),
    *kcak(31),kdakm(3,31),kdak(31),icakm(3,31,20),icak(31,20),
    *idakm(3,31,20),idak(31,20)
    integer*2 izdelm(3),kcdelm(3),kddelm(3),icdelm(3,20),
    *iddelm(3,20),izym(50),kcym(50),kdym(50),icym(50,20),idym(50,20), */
    int *izrab,*kcrab,*kdrab,iz2,
    *icrab,*idrab,*izy,*kcy,*kdy,

    *icy,*idy,*izx,*kcx,*kdx,*pigau,*pizrez,*pkcrez,*pkdrez,

```

```

*pizrl,*pkcl,*pkdl,

*icx,*idx,*izakm,*izak,*kcakm,
*kcak,*kdakm,*kdak,*icakm,*icak,
*idakm,*idak;
int *izdelm,*kcdelm,*kddelm,*icdelm,
*iddelm,*izym,*kcym,*kdym,*icym,*idym,k31,kakm,
icw[400],idw[400],icr[400],idr[400],icrez[400],idrez[400],
icl[400],
idl[400],icn[400],idn[400],icxmi[400],idxmi[400],icdelt[400],
iddelt[400], *izwkl,*kcwkl,*kdwkl,*icwkl, *idwkl;
//izwkl[11],kcwkl[11],kdwkl[11],icwkl[11][20],
// idwkl[10][21]
int icsy[400],idsy[400],icwtot[400],idwtot[400];
double rwwod;long int icrww,idrww,nt;
int knaz,i,j,n,m,l,ios,izn,kcn,kdn,lt,kcel,kdel,k,kna,karg,
inay,ikoy,kmax,krazr,mis,ipy,jl,ila,iarg,narg,km1,k1,in,ik,
isp,ist,izw,kcw,kdw,izr,kcr,kdr,istep,isw,izrez,izrl,kcl,kdl,kcrez,
kdrez,

igau,izxmi,kcxmi,kdxmi,imin,kdro,kprod,iprod,ipech,km,kr,izl,iz,
izsy,kcsy,kdsy,k50,izdelt,kcdelt,kddelt,izwtot,kcwtot,kdwtot;
// k50=10;
kakm=k31=100;
fp6=fopen("dlmnkbc","w");
izrab=(int *)farmalloc(10300*12);
if(izrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR izrab \n");
return;
}
kcrab=(int *)farmalloc(10300*12);
if(kcrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcrab \n");
return;
}
kdrab=(int *)farmalloc(10300*12);
if(kdrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdrab \n");
return;
}
icrab=(int *)farmalloc(215000*12);
if(icrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR icrab \n");
return;
}
idrab=(int *)farmalloc(215000*12);
if(idrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR idrab \n");
return;
}
izy=(int *)farmalloc(101*12);
if(izy==NULL)
{
fprintf(fp6,"NO MEMORY FOR izy \n");
return;
}
kcy=(int *)farmalloc(101*12);

```

```
if(kcy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcy \n");
return;
}
kdy =(int *)farmalloc(101*12);
if(kdy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdy \n");
return;
}
icy =(int *)farmalloc(2200*12);
if(icy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icy \n");
return;
}
idy =(int *)farmalloc(2200*12);
if(idy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idy \n");
return;
}
izx =(int *)farmalloc(2200*12);
if(izx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izx \n");
return;
}
kcx =(int *)farmalloc(2200*12);
if(kcx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcx \n");
return;
}
kdx =(int *)farmalloc(2200*12);
if(kdx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdx \n");
return;
}
icx =(int *)farmalloc(45000*12);
if(icx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icx \n");
return;
}
idx =(int *)farmalloc(45000*12);
if(idx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idx \n");
return;
}
izakm =(int *)farmalloc(510*12);
if(izakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izakm \n");
return;
}
izak =(int *)farmalloc(101*12);
if(izak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izak \n");
```

```
return;
}
kcakm =(int *)farmalloc(510*12);
if(kcakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcakm \n");
return;
}
kcak =(int *)farmalloc(101*12);

if(kcak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcak \n");
return;
}
kdakm =(int *)farmalloc(510*12);
if(kdakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdakm \n");
return;
}
kdak =(int *)farmalloc(101*12);
if(kdak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdak \n");
return;
}
icakm =(int *)farmalloc(11000*12);
if(icakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icakm \n");
return;
}
icak =(int *)farmalloc(2200*12);
if(icak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icak \n");
return;
}
idakm =(int *)farmalloc(11000*12);
if(idakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idakm \n");
return;
}
idak =(int *)farmalloc(2200*12);
if(idak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idak \n");
return;
}
izdelm =(int *)farmalloc(5*12);
if(izdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izdelm \n");
return;
}
kcdelm =(int *)farmalloc(5*12);
if(kcdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcdelm \n");
return;
}
```

```

}

kddelm =(int *)farmalloc(5*12);
if(kddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kddelm \n");
return;
}
icdelm =(int *)farmalloc(110*12);
if(icdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icdelm \n");
return;
}
iddelm =(int *)farmalloc(110*12);
if(iddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR iddelm \n");
return;
}
izym =(int *)farmalloc(101*12);
if(izym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izym \n");
return;
}
kcym =(int *)farmalloc(101*12);

if(kcym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcym \n");
return;
}
kdym =(int *)farmalloc(101*12);
if(kdym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdym \n");
return;
}
icym =(int *)farmalloc(2200*12);

if(icym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icym \n");
return;
}
idym =(int *)farmalloc(2200*12);
if(idym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idym \n");
return;
}
izwkl =(int *)farmalloc(21*12);
if(izwkl ==NULL)
{

fprintf(fp6,"NO MEMORY FOR izwkl \n");
return;
}
kcwkl =(int *)farmalloc(21*12);
if(kcwkl ==NULL)
{

```

```

    fprintf(fp6,"NO MEMORY FOR kcwkl \n");
    return;
}
kdwkl =(int *)farmalloc(21*12);
if(kdwkl ==NULL)
{
    fprintf(fp6,"NO MEMORY FOR kdwkl \n");
    return;
}
icwkl =(int *)farmalloc(500*12);
if(icwkl ==NULL)
{
    fprintf(fp6,"NO MEMORY FOR icwkl \n");
    return;
}
idwkl =(int *)farmalloc(500*12);
if(idwkl ==NULL)
{
    fprintf(fp6,"NO MEMORY FOR idwkl \n");
    return;
}

//    fprintf(fp6,"\nmemory\n");
fp4=fopen("sleddlc","r");
fp7=fopen("name","r");
fp5=fopen("prowcdl2","w");
//    open(4,file='sleddl',status='old')
//    open(6,file='dlmnk2',status='Unknown')
//    open(5,file='prowfdl2',status='Unknown')
//    open(7,file='name',status='old')
//c    open(9,file='prompr2',status='Unknown')
//    read(7,*) knaz
fscanf(fp7,"%d",&knaz);

if(knaz>100) goto m1762;
for(i=1;i<=knaz;i++)
{
    for(j=1;j<=60;j++)
    anaz[i][j]=' ';
    for(j=1;j<=60;j++)
    {
        fscanf(fp7,"%c",&anaz[i][j]);
//        fprintf(fp6,"%c",anaz[i][j]);
    }
//    fprintf(fp6,"\n");
}
//    return;
for(j=1;j<=80;j++)
{

    fscanf(fp4,"%c",&zgl[j]);
    fprintf(fp6,"%c",zgl[j]);
}
    fprintf(fp6,"\n");
    fscanf(fp4,"%d%d",&n,&m);
//    printf("\nВведите количество параметров-аргументов<=10\n");
    fscanf(fp4,"%d",&karg);
    if(karg<=20) goto arg1;
    fprintf(fp6,"\nколичество параметров-аргументов должно быть<=20!\n");
    return;
arg1:;

```

```

// printf("введите через пробел номера аргументов\n");
// for(i=1;i<=karg;i++)
//   fscanf(fp4,"%d",&mnarg[i]);
// printf("Введите через пробел начальный и конечный номер\n");
//   printf("зависимой переменной\n");
//   fscanf(fp4,"%d%d",&inay,&ikoу);
// printf("\nвведите степень полинома<=3\n");
//   fscanf(fp4,"%d",&kmax);
//   if(kmax<=4) goto poll;
//   fprintf(fp6,"степень полинома должна быть<=4\n");
//   return;
poll:;
// printf("\nвведите количество разрядов для арифметики<=20\n");
//   fscanf(fp4,"%d%d",&krazr,&k50);

//   fprintf(fp6,"\n n=%d m=%d\n",n,m);
//   return;
//   if(n<=100) goto m1443;
//   fprintf(fp6,"число строчек-наблюдений должно быть<=100\n");
//   return;
m1443:;
//   if(m<=100) goto m1444;
m1762:;
//   fprintf(fp6,"число параметров-столбиков должно быть<=100\n");
//   return;
m1444:;
//   nt=n;
//   l=0;
m11601:;
//   ios=nt-nt/10*10;
//   l=l+1;
//   id1[l]=ios;
//   nt=nt/10;
//   if(nt>0) goto m11601;
//   izn=1;
//   kcn=1;
//   kdn=1;
//   lt=l;
//   for(l=1;l<=kcn;l++)
//   {
//     icn[l]=id1[lt];
//     lt=lt-1;
//   }
//   idn[l]=0;
//с-ввод исходной матрицы в длинном формате
//   for(i=1;i<=n;i++)
//   {
//     for(j=1;j<=m;j++)
//     {
//       fprintf(fp6,"\ni=%d j=%d\n",i,j);
//       fscanf(fp4,"%lf",&rwwod);
//       icrww=fabs(rwwod);
//       idrww=fabs((fabs(rwwod)-icrww)*pow(10.,9));
//       izr=1;
//       if(rwwod>=0) goto prel;
//       izr=-1;
//     }
//   }
prel:;
//   nt=icrww;
//   l=0;
mprel:;
//   ios=nt-nt/10*10;
//   l=l+1;
//   id1[l]=ios;

```

```

nt=nt/10;
if (nt>0) goto mpre1;
izrab[(i-1)*m+j]=izr;
kcrab[(i-1)*m+j]=1;
kcel=1;
lt=1;
iz1=1;
for (l=1;l<=iz1;l++)
{
icrab[((i-1)*m+j-1)*k50+1]=idl[lt];
lt=lt-1;
}

nt=idrww;
l=0;
mpre2:;
ios=nt-nt/10*10;
l=l+1;
idl[l]=ios;
nt=nt/10;
if (nt>0) goto mpre2;
if (l==9) goto u1;
kcl=9-l;
for (jl=1;jl<=kcl;jl++)
{
l=l+1;
idl[l]=0;
}

u1:;
kdrab[(i-1)*m+j]=1;
lt=1;
iz1=1;
kdel=1;
for (l=1;l<=iz1;l++)
{
idrab[((i-1)*m+j-1)*k50+1]=idl[lt];
lt=lt-1;
}

// write(6,*) izrab(i,j),kcrab(i,j),kdrab(i,j)
// fprintf(fp6,"\n%d %d %d\n",izr,
//         kcel,kdel);
// return;
//     izrab[(i-1)*m+j]=izr;
// kcrab[(i-1)*m+j]=kcel;
// kdrab[(i-1)*m+j]=kdel;
// for (l=1;l<=kcel;l++)
// {
// fscanf(fp4,"%d",&izr);
// fprintf(fp6,"%d",izr);
// icrab[((i-1)*m+j-1)*k50+1]=izr;
// }
// return;
// fprintf(fp6,"\n");
// for (l=1;l<=kdel;l++)
// {
// fscanf(fp4,"%d",&izr);
// fprintf(fp6,"%d",izr);
// idrab[((i-1)*m+j-1)*k50+1]=izr;
// }

```

```

//      fprintf(fp6, "\n");
//      return;
//      read(4, 11002) (idrab(i, j, l), l=1, kdel)
//      fprintf(fp6, "\nцелая часть\n");
//      for(l=1; l<=kcel; l++)
//      fprintf(fp6, "%d", icrab((i-1)*m+j-1)*k50+1);
//      fprintf(fp6, "\n");
//      return;
//      fprintf(fp6, "\ndробная часть\n");
//      for(l=1; l<=kdel; l++)
//      fprintf(fp6, "%d", idrab((i-1)*m+j-1)*k50+1);
//      fprintf(fp6, "\n");
//      return;
//      }
//      }
//1002      continue
//7775      continue
//      return;
//      kna=1;
//      k=kna;
//      fprintf(fp6, "\nстепень полинома=%d\n", kmax);
//      mis=m;
//      m=karg;
//      return;
//      do 2001 ipy=inay, ikoу
//      for(ipy=inay; ipy<=ikoу; ipy++)
//      {
//      k=kna;
//      write(6, 2003) ipy, (anaz(ipy, jl), jl=1, 60)
//      fprintf(fp6, "\nзависимый параметр-%d\n", ipy);
//      for(jl=1; jl<=60; jl++)
//      fprintf(fp6, "%c", anaz[ipy][jl]);
//      fprintf(fp6, "\nномера аргументов\n");
//      for(i=1; i<=karg; i++)
//      fprintf(fp6, "%d ", mnarg[i]);
//      fprintf(fp6, "\n");
m403:;
//с-формирование вектора у для длинной арифметики
//      do 2002 i=1, n
//      for(i=1; i<=n; i++)
//      {
//      izy[i]=izrab((i-1)*mis+ipy);
//      kcy[i]=kcrab((i-1)*mis+ipy);
//      kdy[i]=kdrab((i-1)*mis+ipy);
//      kcel=kcy[i];
//      kdel=kdy[i];
//      fprintf(fp6, "\n izy=%d kcy=%d kdy=%d\n", izy[i], kcel, kdel);
//      fprintf(fp6, "\nцелая часть в главной\n");
//      for(l=1; l<=kcel; l++)
//      {
//      icy[(i-1)*k50+1]=icrab((i-1)*mis+ipy-1)*k50+1);
//      fprintf(fp6, "%d", icy[(i-1)*k50+1]);
//      }
//      fprintf(fp6, "\ndробная часть в главной\n");
//      for(l=1; l<=kdel; l++)
//      {
//      idy[(i-1)*k50+1]=idrab((i-1)*mis+ipy-1)*k50+1);
//      fprintf(fp6, "%d", idy[(i-1)*k50+1]);
//      }
//      }
//с      y(i)=rab(i, ipy)
//2002      continue
//      printf("\n после y\n");

```

```

//с-формирование матрицы x аргументов для длинной арифметики
    ila=0;
    for(iarg=1;iarg<=karg;iarg++)
    {
        narg=mnarg[iarg];
        ila=ila+1;
// 9002 i=1,n
        for(i=1;i<=n;i++)
        {
            izx[(i-1)*m+ila]=izrab[(i-1)*mis+narg];
            kcx[(i-1)*m+ila]=kcrab[(i-1)*mis+narg];
            kdx[(i-1)*m+ila]=kdrab[(i-1)*mis+narg];
            kcel=kcx[(i-1)*m+ila];
            kdel=kdx[(i-1)*m+ila];
            for(l=1;l<=kcel;l++)
                icx[((i-1)*m+ila-1)*k50+1]=icrab[((i-1)*mis+narg-1)*k50+1];

            for(l=1;l<=kdel;l++)
                idx[((i-1)*m+ila-1)*k50+1]=idrab[((i-1)*mis+narg-1)*k50+1];

//c x(i,ila)=rab(i,narg)
//9002 continue
        }// end i
//9001 continue
    } //end iarg
// printf("\npered mnk\n");
// return;

    mnk(n,karg,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
    izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,&igau,krazr,k50);
// fprintf(fp6,"\nposle mnk\n");
// if(k==2) return;
// printf("\n posle mnk\n");
// return;
// printf("\n igau=%d\n",igau);
// fprintf(fp6,"posle mnk igau=%d\n",igau);
// if(igau==0) goto m401;

    km1=k*m+1;
// kakm=100;
// do 8 i=1,km1
    for(i=1;i<=km1;i++)
    {
        izakm[(k-1)*kakm+i]=izak[i];
        kcakm[(k-1)*kakm+i]=kcak[i];
        kdakm[(k-1)*kakm+i]=kdak[i];
        kcel=kcak[i];
        kdel=kdak[i];
        for(l=1;l<=kcel;l++)
            icakm[((k-1)*kakm+i-1)*k50+1]=icak[(i-1)*k50+1];
        for(l=1;l<=kdel;l++)
            idakm[((k-1)*kakm+i-1)*k50+1]=idak[(i-1)*k50+1];
    }
// printf("\n do 8\n");
//с-расчет модельных значений ум(i)
    k1=k+1;
    izdelm[k]=1;
    kcdelm[k]=1;
    kddelm[k]=1;
    icdelm[(k-1)*k50+1]=0;
    iddelm[(k-1)*k50+1]=0;

```

```

//c  delm(k)=0
//  do 54 i=1,n
    for (i=1;i<=n;i++)
    {
        izym[i]=izak[km1];
        kcy[m][i]=kcak[km1];
        kdym[i]=kdak[km1];
        kcel=kcak[km1];
        kdel=kdak[km1];
        for (l=1;l<=kcel;l++)
            icym[(i-1)*k50+l]=icak[(km1-1)*k50+l];
        for (l=1;l<=kdel;l++)
            idym[(i-1)*k50+l]=idak[(km1-1)*k50+l];

//c  ym(i)=ak(km1)
//  do 540 j=1,m
    for (j=1;j<=m;j++)
    {
        in=(j-1)*(k)+1;
        ik=in+k-1;
        isp=1;
//  do 55 ist=in,ik
        for (ist=in;ist<=ik;ist++)
        {
//c  ym(i)=ym(i)+ak(ist)*x(i,j)**(k-(isp-1))
            izw=1;
            kcw=1;
            kdw=1;
            icw[1]=1;
            idw[1]=0;
            izr=izx[(i-1)*m+j];
            kcr=kcx[(i-1)*m+j];
            kdr=kdx[(i-1)*m+j];
            for (l=1;l<=kcr;l++)
                icr[l]=icx[((i-1)*m+j-1)*k50+l];
            for (l=1;l<=kdr;l++)
                idr[l]=idx[((i-1)*m+j-1)*k50+l];
            istep=k-(isp-1);
            if(istep==0) goto m11103;

//  do 11104 isw=1,istep
            for (isw=1;isw<=istep;isw++)
            {
                ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
                    kcr,kdr,icr,idr,
                    &kcrez,&kdreiz,icrez,idrez,krazr);
                izw=izrez;
                kcw=kcrez;
                kdw=kdrez;
                for (l=1;l<=kcw;l++)
                    icw[l]=icrez[l];
                for (l=1;l<=kdw;l++)
                    idw[l]=idrez[l];
            }
m11103:;
            izr=izak[ist];
            kcr=kcak[ist];
            kdr=kdak[ist];
            for (l=1;l<=kcr;l++)
                icr[l]=icak[(ist-1)*k50+l];
            for (l=1;l<=kdr;l++)
                idr[l]=idak[(ist-1)*k50+l];
            ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,

```

```

kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
  izr=izym[i];
  kcr=kcym[i];
  kdr=kdym[i];
  for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
  for(l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];
  slzn(izr,izrez,&izr1,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
  izym[i]=izr1;
  kcym[i]=kcl;
  kdym[i]=kdl;
  for(l=1;l<=kcl;l++)
    icym[(i-1)*k50+1]=icl[l];
  for(l=1;l<=kdl;l++)
    idym[(i-1)*k50+1]=idl[l];
  isp=isp+1;
//55 continue
}

//540      continue
}
//      printf("\n 540\n");
//c      pause '540'
//c      delm(k)=delm(k)+dabs(ym(i)-y(i))
  izr=izym[i];
  kcr=kcym[i];
  kdr=kdym[i];
  for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
  for(l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];

  izr1=izy[i];
  kcl=ky[i];
  kdl=ky[i];
  for(l=1;l<=kcl;l++)
    icl[l]=icy[(i-1)*k50+1];
  for(l=1;l<=kdl;l++)
    idl[l]=idy[(i-1)*k50+1];
  wiczn(izr,izr1,&izrez,kcr,kdr,icr,idr,

kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
  izrez=izrez*izrez;
  izr=izdelm[k];
  kcr=kcdelm[k];
  kdr=kddelm[k];
  for(l=1;l<=kcr;l++)
    icr[l]=icdelm[(k-1)*k50+1];
  for(l=1;l<=kdr;l++)
    idr[l]=iddelm[(k-1)*k50+1];
  slzn(izr,izrez,&izr1,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
  izdelm[k]=izr1;
  kcdelm[k]=kcl;
  kddelm[k]=kdl;
  for(l=1;l<=kcl;l++)
    icdelm[(k-1)*k50+1]=icl[l];

```

```

        for (l=1;l<=kdl;l++)
        iddelm[(k-1)*k50+1]=idl[l];
//54 continue

    }
//    printf("\n 54\n");
//c pause '54'
//c delm(k)=delm(k)/n
    izr=izdelm[k];
    kcr=kcdelm[k];
    kdr=kddelm[k];
    for (l=1;l<=kcr;l++)
    icr[l]=icdelm[(k-1)*k50+1];
    for (l=1;l<=kdr;l++)
    idr[l]=iddelm[(k-1)*k50+1];
    delzn(izr, izn, &izrez, kcr, kdr, icr, idr,
kcn, kdn, icn, idn,
&kcrez, &kdrez, icrez, idrez, krazr);
    izdelm[k]=izrez;
    kcdelm[k]=kcrez;
    kddelm[k]=kdrez;
    for (l=1;l<=kcrez;l++)
    icdelm[(k-1)*k50+1]=icrez[l];
    for (l=1;l<=kdrez;l++)
    iddelm[(k-1)*k50+1]=idrez[l];
    k=k+1;
    if(k<=kmax) goto m403;
//401 continue
m401:;
//    printf("\n m401\n");
//c pause '401'
    if((igau==0)&&(k==1)) goto m9988;
    if(k<kmax) kmax=k-1;
    if((k==kmax)&&(igau==0)) kmax=k-1;
    izxmi=izdelm[kna];
    kcxmi=kcdelm[kna];
    kdxmi=kddelm[kna];
    for (l=1;l<=kcxmi;l++)
    icxmi[l]=icdelm[(kna-1)*k50+1];
    for (l=1;l<=kdxmi;l++)
    idxmi[l]=iddelm[(kna-1)*k50+1];
//c xmin=delm(kna)
    imin=kna;

// do 405 i=kna,kmax
    for (i=kna;i<=kmax;i++)
    {

        kcel=kcdelm[i];
        kdel=kddelm[i];
        if(kcel>kcxmi) goto m405;
        if(kcel<kcxmi) goto m12101;
        for (l=1;l<=kcel;l++)
        icr[l]=icdelm[(i-1)*k50+1];
        for (l=1;l<=kdel;l++)
        idr[l]=iddelm[(i-1)*k50+1];
        for (l=1;l<=kcel;l++)
        {
            if(icr[l]>icxmi[l]) goto m405;
            if(icr[l]<icxmi[l]) goto m12101;
        }
    }

```

```

    kdro=kdel;
    if(kdel>kdxmi) kdro=kdxmi;
    for(l=1;l<=kdro;l++)
    {
        if(idr[l]>idxmi[l]) goto m405;
        if(idr[l]<idxmi[l]) goto m12101;
    }

    if(kdel>=kdxmi) goto m405;

//c  if(kdel.eq.kdxmi) goto 405

//c  if(delm(i).ge.xmin) goto 405
m12101:;
//c  xmin=delm(i)
    izxmi=izdelm[i];
    kcxmi=kcdelm[i];
    kdxmi=kddelm[i];
    for(l=1;l<=kcxmi;l++)
        icxmi[l]=icdelm[(i-1)*k50+1];

    for(l=1;l<=kdxmi;l++)
        idxmi[l]=iddelm[(i-1)*k50+1];
    imin=i;
m405:;
    }

//с-проверить необходимость метки 406
//406      continue
//с  pause '406'
//с-расчет модельных значений um(i) по min ошибке
    fprintf(fp6, "\ноптимальная степень=%d\n", imin);

    fprintf(fp6, "\nМодель для зависимого параметра\n");
    kprod=7;
    iprod=0;
    ipech=0;
//  do 4001 iarg=1,karg
    for(iarg=1;iarg<=karg;iarg++)
    {

        narg=mnarg[iarg];
        fprintf(fp6, "\наргумент-%d\n", narg);
        for(jl=1;jl<=60;jl++)
            fprintf(fp6, "%c", anaz[narg][jl]);

        km1=imin*m+1;
        km=imin*m;
        kr=imin;

        in=(iarg-1)*imin+1;
        ik=in+imin-1;

//  do 876 j=in,ik
        for(j=in;j<=ik;j++)
        {

//  write(6,12451) narg,kr
        fprintf(fp6, "\номер аргумента-%d степень=%d\n", narg, kr);
//      k31=100;
            iz1=izakm[(imin-1)*k31+j];
            kcel=kcakm[(imin-1)*k31+j];

```

```

kdel=kdakm[ (imin-1)*k31+j];
for(l=1;l<=kcel;l++)
icr[l]=icakm[ ((imin-1)*k31+j-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[ ((imin-1)*k31+j-1)*k50+1];

fprintf(fp6, "\nзнак=%d\n", iz1);

fprintf(fp6, "\nцелая часть коэффициента\n");
for(l=1;l<=kcel;l++)
fprintf(fp6, "%1d", icr[l]);
fprintf(fp6, "\n");
fprintf(fp6, "\ndробная часть коэффициента\n");
for(l=1;l<=kdel;l++)
fprintf(fp6, "%1d", idr[l]);
fprintf(fp6, "\n");
//c write(9,50211) akm(imin,j),narg,kr
//c write(6,50211) akm(imin,j),narg,kr

kr=kr-1;
m876:;
}
m4001:;
}
//c write(6,14007) akm(imin,km1)
//c write(9,14007) akm(imin,km1)
//c write(*,14007) akm(imin,km1)
//c pause
iz=izakm[ (imin-1)*k31+km1];
kcel=kcakm[ (imin-1)*k31+km1];
kdel=kdakm[ (imin-1)*k31+km1];
for(l=1;l<=kcel;l++)
icr[l]=icakm[ ((imin-1)*k31+km1-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[ ((imin-1)*k31+km1-1)*k50+1];

fprintf(fp6, "\nсвободный член модели\n");

fprintf(fp6, "\nзнак свободного члена=%d\n", iz);

fprintf(fp6, "\nцелая часть свободного члена\n");
for(l=1;l<=kcel;l++)
fprintf(fp6, "%1d", icr[l]);

fprintf(fp6, "\ndробная часть свободного члена\n");
for(l=1;l<=kdel;l++)

fprintf(fp6, "%1d", idr[l]);
//c delt=0
izdelt=1;
kcdelt=1;
kddelt=1;
icdelt[1]=0;
iddelt[1]=0;
// do 5201 i=1,m
for(i=1;i<=m;i++)
{
izwkl[i]=1;
kcwkl[i]=1;
kdwkl[i]=1;

```

```

//c  wkl(i)=0
      icwkl[(i-1)*k50+1]=0;
      idwkl[(i-1)*k50+1]=0;
//5201  continue
      }

      izsy=1;
      kcsy=1;
      kdsy=1;
//c  sy=0
      icsy[1]=0;
      idsy[1]=0;
//  do 3354 i=1,n
      for(i=1;i<=n;i++)
      {

//c  ym(i)=akm(imin,km1)
      izym[i]=izakm[(imin-1)*k31+km1];
      kcym[i]=kcakm[(imin-1)*k31+km1];
      kdym[i]=kdakm[(imin-1)*k31+km1];
      kcel=kcym[i];
      kdel=kdym[i];
      for(l=1;l<=kcel;l++)
      icym[(i-1)*k50+1]=icakm[((imin-1)*k31+km1-1)*k50+1];

      for(l=1;l<=kdel;l++)
      idym[(i-1)*k50+1]=idakm[((imin-1)*k31+km1-1)*k50+1];

//  do 33540 j=1,m
      for(j=1;j<=m;j++)
      {

      in=(j-1)*(imin)+1;
      ik=in+imin-1;
      isp=1;
//  do 3355 ist=in,ik
      for(ist=in;ist<=ik;ist++)
      {

      istep=imin-(isp-1);
      izw=1;
      kcw=1;
      kdw=1;

      icw[1]=1;
      idw[1]=0;
      izr=izx[(i-1)*m+j];
      kcr=kcx[(i-1)*m+j];
      kdr=kdx[(i-1)*m+j];
      for(l=1;l<=kcr;l++)
      icr[l]=icx[((i-1)*m+j-1)*k50+1];

      for(l=1;l<=kdr;l++)
      idr[l]=idx[((i-1)*m+j-1)*k50+1];

      if(istep==0) goto m31103;
//  do 31104 isw=1,istep
      for(isw=1;isw<=istep;isw++)
      {
      ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
      kcr,kdr,icr,idr,
      &kcrez,&kdreiz,icrez,idrez,krazr);
      izw=izrez;

```

```

    kcw=kcrez;
    kdw=kdrez;
    for (l=1;l<=kcw;l++)
    icw[l]=icrez[l];
    for (l=1;l<=kdw;l++)
    idw[l]=idrez[l];

//31104     continue
    }
m31103:;
    izr=izakm[(imin-1)*k31+ist];
    kcr=kcakm[(imin-1)*k31+ist];
    kdr=kdakm[(imin-1)*k31+ist];
    for (l=1;l<=kcr;l++)
    icr[l]=icakm[((imin-1)*k31+ist-1)*k50+1];
    for (l=1;l<=kdr;l++)
    idr[l]=idakm[((imin-1)*k31+ist-1)*k50+1];

    ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
    kcr,kdr,icr,idr,
    &kcrez,&kdrez,icrez,idrez,krazr);
    iz1=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for (l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];

    for (l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];

    slzn(iz1,izrez,&iz2,kcr,kdr,icr,idr,
    kcrez,kdrez,icrez,idrez,
    &kcl,&kdl,icl,idl,krazr);
    izym[i]=iz2;
    kcym[i]=kcl;
    kdym[i]=kdl;
    for (l=1;l<=kcl;l++)
    icym[(i-1)*k50+1]=icl[l];
    for (l=1;l<=kdl;l++)
    idym[(i-1)*k50+1]=idl[l];

//c     ym(i)=ym(i)+akm(imin,ist)*x(i,j)**(imin-(isp-1))
    iz1=izwkl[j];
    kcel=kcwkl[j];

    kdel=kdwkl[j];
    for (l=1;l<=kcel;l++)
    icr[l]=icwkl[(j-1)*k50+1];
    for (l=1;l<=kdel;l++)
    idr[l]=idwkl[(j-1)*k50+1];
    izrez=izrez*izrez;
    slzn(iz1,izrez,&iz2,kcel,kdel,icr,idr,
    kcrez,kdrez,icrez,idrez,
    &kcl,&kdl,icl,idl,krazr);
//c     wk1(j)=wk1(j)+dabs(akm(imin,ist)*x(i,j)**(imin-(isp-1)))

    izwkl[j]=iz2;
    kcwkl[j]=kcl;
    kdwkl[j]=kdl;
    for (l=1;l<=kcl;l++)
    icwkl[(j-1)*k50+1]=icl[l];
    for (l=1;l<=kdl;l++)
    idwkl[(j-1)*k50+1]=idl[l];

```

```

        isp=isp+1;
//3355     continue
    }
//33540    continue
    }
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for(l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];

    izl=izakm[(imin-1)*k31+km1];
    kcl=kcakm[(imin-1)*k31+km1];
    kdl=kdakm[(imin-1)*k31+km1];
    for(l=1;l<=kcl;l++)
    icl[l]=icakm[((imin-1)*k31+km1-1)*k50+1];
    for(l=1;l<=kdl;l++)
    idl[l]=idakm[((imin-1)*k31+km1-1)*k50+1];
    wiczn(izr,izl,&izrez,kcr,kdr,icr,idr,
    kcl,kdl,icl,idl,
//c     sy=sy+y(i)-akm(imin,km1)
        slzn(izsy,izrez,&iz2,kcsy,kdsy,icsy,idsy,
    kcrez,kdrez,icrez,idrez,
    &kcl,&kdl,icl,idl,krazr);
    izsy=iz2;
    kcsy=kcl;
    kdsy=kdl;
    for(l=1;l<=kcl;l++)
    icsy[l]=icl[l];
    for(l=1;l<=kdsy;l++)
    idsy[l]=idl[l];
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for(l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];
/*c     write(6,*) i
c     write(6,*) 'модельное y'
c     write(6,*) izr
c     write(6,*) 'целая часть'
c     write(6,87321) (icr(l),l=1,kcr)
87321 format(1x,60i1)
c     write(6,*) 'дробная часть'
c     write(6,87321) (idr(l),l=1,kdr) */
    izl=izy[i];
    kcl=ky[i];
    kdl=ky[i];
    for(l=1;l<=kcl;l++)
    icl[l]=icy[(i-1)*k50+1];
    for(l=1;l<=kdl;l++)
    idl[l]=idy[(i-1)*k50+1];
/*c     write(6,*) 'исходное y'

c     write(6,*) izl

c     write(6,*) 'целая часть'
c     write(6,87321) (icl(l),l=1,kcl)

```

```

c   write(6,*) 'дробная часть'
c   write(6,87321) (idl(1),l=1,kdl)   */
      wiczn(izr,izl,&iz2,kcr,kdr,icr,idr,
      kcl,kdl,icl,idl,
      &kcrez,&kdrez,icrez,idrez,krazr);
      izrez=iz2*iz2;
//c   delt=delt+dabs(ym(i)-y(i))
      slzn(izdelt,izrez,&iz2,kcdelt,kddelt,icdelt,iddelt,
      kcrez,kdrez,icrez,idrez,
      &kcl,&kdl,icl,idl,krazr);
      izdelt=iz2;
      kcdelt=kcl;
      kddelt=kdl;
      for(l=1;l<=kcl;l++)
      icdelt[l]=icl[l];
      for(l=1;l<=kdl;l++)
      iddelt[l]=idl[l];

//c345      format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
m3354:;
      }
      izwtot=1;
      kcwtot=1;
      kdwtot=1;
      icwtot[1]=0;
      idwtot[1]=0;
//c   wkltot=0
//   do 5204 i=1,m
      for(i=1;i<=m;i++)
      {
      izr=1;
      kcr=kcwkl[i];
      kdr=kdwkl[i];
      for(l=1;l<=kcr;l++)
      icr[l]=icwkl[(i-1)*k50+1];
      for(l=1;l<=kdr;l++)
      idr[l]=idwkl[(i-1)*k50+1];
//c   wkltot=wkltot+dabs(wkl(i))
      slzn(izwtot,izr,&izl,kcwtot,kdwtot,icwtot,idwtot,
      kcr,kdr,icr,idr,
      &kcrez,&kdrez,icrez,idrez,krazr);
      izwtot=izl;
      kcwtot=kcrez;
      kdwtot=kdrez;
      for(l=1;l<=kcrez;l++)
      icwtot[l]=icrez[l];
      for(l=1;l<=kdrez;l++)
      idwtot[l]=idrez[l];
//5204      continue
      }
      for(i=1;i<=m;i++)
      {
      izr=izwkl[i];
      kcr=kcwkl[i];
      kdr=kdwkl[i];
      for(l=1;l<=kcr;l++)
      icr[l]=icwkl[(i-1)*k50+1];
      for(l=1;l<=kdr;l++)
      idr[l]=idwkl[(i-1)*k50+1];
//c   wkl(i)=wkl(i)/wkltot
      delzn(izr,izwtot,&izl,kcr,kdr,icr,idr,
      kcwtot,kdwtot,icwtot,idwtot,
      &kcl,&kdl,icl,idl,krazr);

```

```

        izwkl[i]=izl;
        kcwkl[i]=kc1;
        kdwkl[i]=kd1;
        for (l=1;l<=kc1;l++)
            icwkl[(i-1)*k50+l]=ic1[l];
        for (l=1;l<=kd1;l++)
            idwkl[(i-1)*k50+l]=id1[l];
//5203         continue
//5202         continue
    }
//c    delt=delt/n
    delzn(izdelt,izn,&izr,kcdelt,kddelt,icdelt,iddelt,
    kcn,kdn,icn,idn,
    &kcrez,&kdrez,icrez,idrez,krazr);
    izdelt=izr;
    kcdelt=kcrez;
    kddelt=kdrez;
    for (l=1;l<=kcrez;l++)
        icdelt[l]=icrez[l];
    for (l=1;l<=kdrez;l++)
        iddelt[l]=idrez[l];

    fprintf(fp6,"\nсредняя абсолютная ошибка\n");
    fprintf(fp6,"\nцелая часть ошибки\n");
    for (l=1;l<=kcdelt;l++)
        fprintf(fp6,"%ld",icdelt[l]);
    fprintf(fp6,"\ndробная часть ошибки\n");
    for (l=1;l<=kddelt;l++)
        fprintf(fp6,"%ld",iddelt[l]);

    fprintf(fp6,"\n");
//    prob=' '
//    do 5404 i=1,m
    for (i=1;i<=m;i++)
    {
        narg=mnarg[i];
//    write(6,5405) narg,(anaz(narg,j),j=1,60)
        fprintf(fp6,"\nномер аргумента=%d\n",narg);
        for (j=1;j<=60;j++)
            fprintf(fp6,"%c",anaz[narg][j]);
            fprintf(fp6,"\nцелая часть вклада\n");
        kcel=kcwkl[i];
        for (l=1;l<=kcel;l++)
            fprintf(fp6,"%ld",icwkl[(i-1)*k50+l]);
        fprintf(fp6,"\ndробная часть вклада\n");
        kdel=kdwkl[i];
        for (l=1;l<=kdel;l++)
            fprintf(fp6,"%ld",idwkl[(i-1)*k50+l]);
            fprintf(fp6,"\n");
//5404         continue
    }
//891 format(/lx,'средняя абсолютная ошибка=',f36.18/)
m9988:;
m2001:;
    }
    fclose(fp4);
    fclose(fp6);
    fclose(fp5);
    fclose(fp7);
    return;
}

```

## Набор dlmncraz.cpp(поразрядный ввод целой и дробной частей)

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlmnkbc.h"
#include<math.h>
#include<alloc.h>

//      FILE *fp4,*fp6,*fp5,*fp7,*fp9;

void dlmnkbc()
{
extern FILE *fp4,*fp6,*fp5,*fp7,*fp9;

extern void mnk(int n,int karg,int izn,
    int kcn,int kdn,int *icn,int *idn,int *izx,int *kcx,int *kdx,
    int *icx,int *idx,
    int *izy,int *kcy,int *kdy,int *icy,int *idy,int k,
    int *izak,int *kcak,int *kdak,
    int *icak,int *idak,int *pigau,int krazr,int k50);

    extern void ymnzn(int izw,int izr,int *pizrez,
        int kcw,int kdw,int *icw,int *idw,
        int kcr,int kdr,int *icr,int *idr,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void wiczn(int izr,int izrl,int *pizrez,
        int kcr,int kdr,int *icr,int *idr,

        int kcl,int kdl,int *icl,int *idl,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void slzn(int izr,int izrez,int *pizrl,
        int kcr,int kdr,int *icr,int *idr,
        int kcrez,int kdrez,int *icrez,int *idrez,
        int *pkcl,int *pkdl,int *icl,int *idl,int krazr);

    extern void delzn(int izr,int izn,int *pizrez,
        int kcr,int kdr,int *icr,int *idr,
        int kcn,int kdn,int *icn,int *idn,
        int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

//с-метод наименьших квадратов в длинной арифметике
    char anaz[101][61];int mnarg[21];char zgl[81];
/*
    integer*2 izrab(50,5),kcrab(50,20),kdrab(50,20),
    *icrab(50,20,20),idrab(50,20,20),izy(50),kcy(50),kdy(50),

    *icy(50,20),idy(50,20),izx(50,10),kcx(50,10),kdx(50,10),
    *icx(50,10,20),idx(50,10,20),izakm(3,31),izak(31),kcakm(3,31),
    *kcak(31),kdakm(3,31),kdak(31),icakm(3,31,20),icak(31,20),
    *idakm(3,31,20),idak(31,20)
    integer*2 izdelm(3),kcdelm(3),kddelm(3),icdelm(3,20),
    *iddelm(3,20),izym(50),kcy(50),kdym(50),icy(50,20),idy(50,20), */
    int *izrab,*kcrab,*kdrab,iz2,
    *icrab,*idrab,*izy,*kcy,*kdy,

    *icy,*idy,*izx,*kcx,*kdx,*pigau,*pizrez,*pkcrez,*pkdrez,

```

```

*pizrl,*pkcl,*pkdl,

*icx,*idx,*izakm,*izak,*kcakm,
*kcak,*kdakm,*kdak,*icakm,*icak,
*idakm,*idak;
int *izdelm,*kcdelm,*kddelm,*icdelm,
*iddelm,*izym,*kcym,*kdym,*icym,*idym,k31,kakm,
icw[400],idw[400],icr[400],idr[400],icrez[400],idrez[400],
icl[400],
idl[400],icn[400],idn[400],icxmi[400],idxmi[400],icdelt[400],
iddelt[400], *izwkl,*kcwkl,*kdwkl,*icwkl, *idwkl;
//izwkl[11],kcwkl[11],kdwkl[11],icwkl[11][20],
// idwkl[10][21]
int icsy[400],idsy[400],icwtot[400],idwtot[400];
double rwwod;long int icrww,idrww,nt;
int knaz,i,j,n,m,l,ios,izn,kcn,kdn,lt,kcel,kdel,k,kna,karg,
inay,ikoy,kmax,krazr,mis,ipy,jl,ila,iarg,narg,km1,k1,in,ik,
isp,ist,izw,kcw,kdw,izr,kcr,kdr,istep,isw,izrez,izrl,kcl,kdl,kcrez,
kdrez,

igau,izxmi,kcxmi,kdxmi,imin,kdro,kprod,iprod,ipech,km,kr,izl,iz,
izsy,kcsy,kdsy,k50,izdelt,kcdelt,kddelt,izwtot,kcwtot,kdwtot;
// k50=10;
kakm=k31=100;
fp6=fopen("dlmncraz","w");
izrab=(int *)farmalloc(10300*12);
if(izrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR izrab \n");
return;
}
kcrab=(int *)farmalloc(10300*12);
if(kcrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcrab \n");
return;
}
kdrab=(int *)farmalloc(10300*12);
if(kdrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdrab \n");
return;
}
icrab=(int *)farmalloc(215000*12);
if(icrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR icrab \n");
return;
}
idrab=(int *)farmalloc(215000*12);
if(idrab==NULL)
{
fprintf(fp6,"NO MEMORY FOR idrab \n");
return;
}
izy=(int *)farmalloc(101*12);
if(izy==NULL)
{
fprintf(fp6,"NO MEMORY FOR izy \n");
return;
}
kcy=(int *)farmalloc(101*12);

```

```
if(kcy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcy \n");
return;
}
kdy =(int *)farmalloc(101*12);
if(kdy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdy \n");
return;
}
icy =(int *)farmalloc(2200*12);
if(icy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icy \n");
return;
}
idy =(int *)farmalloc(2200*12);
if(idy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idy \n");
return;
}
izx =(int *)farmalloc(2200*12);
if(izx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izx \n");
return;
}
kcx =(int *)farmalloc(2200*12);
if(kcx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcx \n");
return;
}
kdx =(int *)farmalloc(2200*12);
if(kdx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdx \n");
return;
}
icx =(int *)farmalloc(45000*12);
if(icx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icx \n");
return;
}
idx =(int *)farmalloc(45000*12);
if(idx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idx \n");
return;
}
izakm =(int *)farmalloc(510*12);
if(izakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izakm \n");
return;
}
izak =(int *)farmalloc(101*12);
if(izak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izak \n");
```

```
return;
}
kcakm =(int *)farmalloc(510*12);
if(kcakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcakm \n");
return;
}
kcak =(int *)farmalloc(101*12);
if(kcak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcak \n");
return;
}
kdakm =(int *)farmalloc(510*12);
if(kdakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdakm \n");
return;
}
kdak =(int *)farmalloc(101*12);
if(kdak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdak \n");
return;
}
icakm =(int *)farmalloc(11000*12);
if(icakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icakm \n");
return;
}
icak =(int *)farmalloc(2200*12);
if(icak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icak \n");
return;
}
idakm =(int *)farmalloc(11000*12);
if(idakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idakm \n");
return;
}
idak =(int *)farmalloc(2200*12);
if(idak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idak \n");
return;
}
izdelm =(int *)farmalloc(5*12);
if(izdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izdelm \n");
return;
}
kcdelm =(int *)farmalloc(5*12);
if(kcdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcdelm \n");
return;
}
kddelm =(int *)farmalloc(5*12);
```

```

if(kddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kddelm \n");
return;
}
icdelm =(int *)farmalloc(110*12);
if(icdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icdelm \n");
return;

}
iddelm =(int *)farmalloc(110*12);
if(iddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR iddelm \n");
return;
}
izym =(int *)farmalloc(101*12);
if(izym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izym \n");
return;
}
kcym =(int *)farmalloc(101*12);
if(kcym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcym \n");
return;
}
kdym =(int *)farmalloc(101*12);
if(kdym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdym \n");
return;
}
icym =(int *)farmalloc(2200*12);

if(icym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icym \n");
return;
}
idym =(int *)farmalloc(2200*12);
if(idym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idym \n");
return;
}
izwkl =(int *)farmalloc(21*12);
if(izwkl ==NULL)
{

fprintf(fp6,"NO MEMORY FOR izwkl \n");
return;
}
kcwkl =(int *)farmalloc(21*12);
if(kcwkl ==NULL)
{

fprintf(fp6,"NO MEMORY FOR kcwkl \n");
return;
}

```

```

kdwk1 =(int *)farmalloc(21*12);
if(kdwk1 ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdwk1 \n");

return;
}
icwk1 =(int *)farmalloc(500*12);
if(icwk1 ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icwk1 \n");
return;
}
idwk1 =(int *)farmalloc(500*12);
if(idwk1 ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idwk1 \n");
return;

}

// fprintf(fp6,"\nmemory\n");
fp4=fopen("sledraz","r");
fp7=fopen("name","r");
fp5=fopen("prowcraz","w");
// open(4,file='sleddl',status='old')
// open(6,file='dlmnk2',status='Unknown')
// open(5,file='prowfdl2',status='Unknown')
// open(7,file='name',status='old')
//c open(9,file='prompr2',status='Unknown')
// read(7,*) knaz
fscanf(fp7,"%d",&knaz);

if(knaz>100) goto m1762;
for(i=1;i<=knaz;i++)
{
for(j=1;j<=60;j++)
anaz[i][j]=' ';
for(j=1;j<=60;j++)
{
fscanf(fp7,"%c",&anaz[i][j]);
fprintf(fp6,"%c",anaz[i][j]);
}
fprintf(fp6,"\n");
}
// return;
for(j=1;j<=80;j++)
{

fscanf(fp4,"%c",&zgl[j]);
fprintf(fp6,"%c",zgl[j]);
}
fprintf(fp6,"\n");
fscanf(fp4,"%d%d",&n,&m);
// printf("\nВведите количество параметров-аргументов<=10\n");
fscanf(fp4,"%d",&karg);
if(karg<=20) goto arg1;
fprintf(fp6,"\nколичество параметров-аргументов должно быть<=20!\n");
return;
arg1:;
// printf("введите через пробел номера аргументов\n");
for(i=1;i<=karg;i++)
fscanf(fp4,"%d",&mnarg[i]);

```

```

// printf("Введите через пробел начальный и конечный номер\n");
// printf("зависимой переменной\n");
fscanf(fp4,"%d%d",&inaу,&ikoу);
// printf("\nвведите степень полинома<=3\n");
fscanf(fp4,"%d",&kмаx);
if(kмаx<=4) goto poll;
fprintf(fp6,"степень полинома должна быть<=4\n");
return;
poll:;
// printf("\nвведите количество разрядов для арифметики<=20\n");
fscanf(fp4,"%d%d",&kразr,&k50);

// fprintf(fp6,"\n n=%d m=%d\n",n,m);
// return;
if(n<=100) goto m1443;
fprintf(fp6,"число строчек-наблюдений должно быть<=100\n");
return;
m1443:;
if(m<=100) goto m1444;
m1762:;
fprintf(fp6,"число параметров-столбиков должно быть<=100\n");
return;
m1444:;
nt=n;
l=0;
m11601:;
ios=nt-nt/10*10;
l=l+1;
id1[l]=ios;
nt=nt/10;
if(nt>0) goto m11601;
izn=1;
kcn=1;
kdn=1;
lt=1;
for(l=1;l<=kcn;l++)
{
icn[l]=id1[lt];
lt=lt-1;
}
idn[1]=0;
//с-ввод исходной матрицы в длинном формате
for(i=1;i<=n;i++)
{
for(j=1;j<=m;j++)
{
fprintf(fp6,"\ni=%d j=%d\n",i,j);
fscanf(fp4,"%d%d%d",&izrab[(i-1)*m+j],&kcrab[(i-1)*m+j],
&kdrab[(i-1)*m+j]);
izr=izrab[(i-1)*m+j];
kcel=kcrab[(i-1)*m+j];
kdel=kdrab[(i-1)*m+j];
// write(6,*) izrab(i,j),kcrab(i,j),kdrab(i,j)
fprintf(fp6,"\n%d %d %d\n",izr,
kcel,kdel);
// return;
// izrab[(i-1)*m+j]=izr;
// kcrab[(i-1)*m+j]=kcel;
// kdrab[(i-1)*m+j]=kdel;
// for(l=1;l<=kcel;l++)
// {
// fscanf(fp4,"%d",&izr);

```

```

//      fprintf(fp6,"%d",izr);
//      icrab[((i-1)*m+j-1)*k50+1]=izr;
//      }
//      return;
//      fprintf(fp6,"\n");
//      for(l=1;l<=kdel;l++)
//      {
//      fscanf(fp4,"%d",&izr);
//      fprintf(fp6,"%d",izr);
//      idrab[((i-1)*m+j-1)*k50+1]=izr;
//      }
//      fprintf(fp6,"\n");
//      return;
//      read(4,11002) (idrab(i,j,l),l=1,kdel)
//      fprintf(fp6,"\nцелая часть\n");
//      for(l=1;l<=kcel;l++)
//      {
//      fscanf(fp4,"%d",&icrab[((i-1)*m+j-1)*k50+1]);
//      fprintf(fp6,"%d",icrab[((i-1)*m+j-1)*k50+1]);
//      }
//      fprintf(fp6,"\n");
//      return;
//      fprintf(fp6,"\nдробная часть\n");
//      for(l=1;l<=kdel;l++)
//      {
//
//      fscanf(fp4,"%d",&idrab[((i-1)*m+j-1)*k50+1]);
//      fprintf(fp6,"%d",idrab[((i-1)*m+j-1)*k50+1]);
//      }
//      fprintf(fp6,"\n");
//      return;
//      }
//      }
//1002      continue
//7775      continue
//      return;
//      kna=1;
//      k=kna;
//      fprintf(fp6,"\nстепень полинома=%d\n",kmax);
//      mis=m;
//      m=karg;
//      return;
//      do 2001 ipy=inay,ikoу
//      for(ipy=inay;ipy<=ikoу;ipy++)
//      {
//      k=kna;
//      write(6,2003) ipy,(anaz(ipy,jl),jl=1,60)
//      fprintf(fp6,"\nзависимый параметр-%d\n",ipy);
//      for(jl=1;jl<=60;jl++)
//      fprintf(fp6,"%c",anaz[ipy][jl]);
//      fprintf(fp6,"\nномера аргументов\n");
//      for(i=1;i<=karg;i++)
//      fprintf(fp6,"%d ",mnarg[i]);
//      fprintf(fp6,"\n");
m403:;
//с-формирование вектора у для длинной арифметики
//      do 2002 i=1,n
//      for(i=1;i<=n;i++)
//      {
//      izy[i]=izrab[(i-1)*mis+ipy];
//      kcy[i]=kcrab[(i-1)*mis+ipy];
//      kdy[i]=kdrab[(i-1)*mis+ipy];
//      kcel=kcy[i];

```

```

    kdel=kdy[i];
//   fprintf(fp6,"\n izy=%d kcy=%d kdy=%d\n",izy[i],kcel,kdel);
//   fprintf(fp6,"целая часть в главной\n");
    for(l=1;l<=kcel;l++)
    {
//   icy[(i-1)*k50+1]=icrab[((i-1)*mis+ipy-1)*k50+1];
//   fprintf(fp6,"%d",icy[(i-1)*k50+1]);
    }
//   fprintf(fp6,"\ndробная часть в главной\n");
    for(l=1;l<=kdel;l++)
    {
//   idy[(i-1)*k50+1]=idrab[((i-1)*mis+ipy-1)*k50+1];
//   fprintf(fp6,"%d",idy[(i-1)*k50+1]);
    }
//c   y(i)=rab(i,ipy)
//2002   continue
//   printf("\n posle y\n");
//c-формирование матрицы x аргументов для длинной арифметики
    ila=0;
    for(iarg=1;iarg<=karg;iarg++)
    {
        narg=mnarg[iarg];
        ila=ila+1;
//   do 9002 i=1,n
        for(i=1;i<=n;i++)
        {
            izx[(i-1)*m+ila]=izrab[(i-1)*mis+narg];
            kcx[(i-1)*m+ila]=kcrab[(i-1)*mis+narg];
            kdx[(i-1)*m+ila]=kdrab[(i-1)*mis+narg];
            kcel=kcx[(i-1)*m+ila];
            kdel=kdx[(i-1)*m+ila];
            for(l=1;l<=kcel;l++)
                icx[((i-1)*m+ila-1)*k50+1]=icrab[((i-1)*mis+narg-1)*k50+1];

            for(l=1;l<=kdel;l++)
                idx[((i-1)*m+ila-1)*k50+1]=idrab[((i-1)*mis+narg-1)*k50+1];

//c   x(i,ila)=rab(i,narg)
//9002   continue
        }// end i
//9001   continue
    } //end iarg
//   printf("\npered mnk\n");
//   return;

    mnk(n,karg,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
    izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,&igau,krazr,k50);
//   fprintf(fp6,"\nposle mnk\n");
//   if(k==2) return;
//   printf("\n posle mnk\n");
//   return;
//   printf("\n igau=%d\n",igau);
//   fprintf(fp6,"posle mnk igau=%d\n",igau);
//   if(igau==0) goto m401;

    kml=k*m+1;
//   kakm=100;
//   do 8 i=1,kml
    for(i=1;i<=kml;i++)
    {
        izakm[(k-1)*kakm+i]=izak[i];
    }

```

```

kcakm[(k-1)*kakm+i]=kcak[i];
kdakm[(k-1)*kakm+i]=kdak[i];
kcel=kcak[i];
kdel=kdak[i];
for(l=1;l<=kcel;l++)
icakm[((k-1)*kakm+i-1)*k50+1]=icak[(i-1)*k50+1];
for(l=1;l<=kdel;l++)
idakm[((k-1)*kakm+i-1)*k50+1]=idak[(i-1)*k50+1];
}
//      printf("\n do 8\n");
//с-расчет модельных значений ym(i)
k1=k+1;
izdelm[k]=1;
kcdelm[k]=1;
kddelm[k]=1;
icdelm[(k-1)*k50+1]=0;
iddelm[(k-1)*k50+1]=0;

//c  delm(k)=0
//  do 54 i=1,n
    for(i=1;i<=n;i++)
    {
        izym[i]=izak[km1];
        kcy[m]=kcak[km1];
        kdym[i]=kdak[km1];
        kcel=kcak[km1];
        kdel=kdak[km1];
        for(l=1;l<=kcel;l++)
        icym[(i-1)*k50+1]=icak[(km1-1)*k50+1];
        for(l=1;l<=kdel;l++)
        idym[(i-1)*k50+1]=idak[(km1-1)*k50+1];

//c  ym(i)=ak(km1)
//  do 540 j=1,m
        for(j=1;j<=m;j++)
        {
            in=(j-1)*(k)+1;
            ik=in+k-1;
            isp=1;
//  do 55 ist=in,ik
            for(ist=in;ist<=ik;ist++)
            {
//c  ym(i)=ym(i)+ak(ist)*x(i,j)**(k-(isp-1))
                izw=1;
                kcw=1;
                kdw=1;
                icw[1]=1;
                idw[1]=0;
                izr=izx[(i-1)*m+j];
                kcr=kcx[(i-1)*m+j];
                kdr=kdx[(i-1)*m+j];
                for(l=1;l<=kcr;l++)
                icr[l]=icx[((i-1)*m+j-1)*k50+1];
                for(l=1;l<=kdr;l++)
                idr[l]=idx[((i-1)*m+j-1)*k50+1];
                istep=k-(isp-1);
                if(istep==0) goto m11103;

//  do 11104 isw=1,istep
                for(isw=1;isw<=istep;isw++)
                {
                    ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
                        kcr,kdr,icr,idr,

```

```

&kcrez, &kdrez, icrez, idrez, krazr);
  izw=izrez;
  kcw=kcrez;
  kdw=kdrez;
  for (l=1; l<=kcw; l++)
    icw[l]=icrez[l];
  for (l=1; l<=kdw; l++)
    idw[l]=idrez[l];
}
m11103;
  izr=izak[ist];
  kcr=kcak[ist];
  kdr=kdak[ist];
  for (l=1; l<=kcr; l++)
    icr[l]=icak[(ist-1)*k50+1];
  for (l=1; l<=kdr; l++)
    idr[l]=idak[(ist-1)*k50+1];
  ymnzn(izw, izr, &izrez, kcw, kdw, icw, idw,
  kcr, kdr, icr, idr,
  &kcrez, &kdrez, icrez, idrez, krazr);
  izr=izym[i];
  kcr=kcym[i];
  kdr=kdym[i];
  for (l=1; l<=kcr; l++)
    icr[l]=icym[(i-1)*k50+1];
  for (l=1; l<=kdr; l++)
    idr[l]=idy[(i-1)*k50+1];
  slzn(izr, izrez, &izr1, kcr, kdr, icr, idr,
  kcrez, kdrez, icrez, idrez,
  &kcl, &kdl, icl, idl, krazr);
  izym[i]=izr1;
  kcym[i]=kcl;
  kdym[i]=kdl;
  for (l=1; l<=kcl; l++)
    icym[(i-1)*k50+1]=icl[l];
  for (l=1; l<=kdl; l++)
    idym[(i-1)*k50+1]=idl[l];
  isp=isp+1;
//55 continue
}

//540 continue
}
// printf("\n 540\n");
//c pause '540'
//c delm(k)=delm(k)+dabs(ym(i)-y(i))
  izr=izym[i];
  kcr=kcym[i];
  kdr=kdym[i];
  for (l=1; l<=kcr; l++)
    icr[l]=icym[(i-1)*k50+1];
  for (l=1; l<=kdr; l++)
    idr[l]=idy[(i-1)*k50+1];

  izr1=izy[i];
  kcl=ky[i];
  kdl=ky[i];
  for (l=1; l<=kcl; l++)
    icl[l]=icy[(i-1)*k50+1];
  for (l=1; l<=kdl; l++)
    idl[l]=idy[(i-1)*k50+1];
  wiczn(izr, izr1, &izrez, kcr, kdr, icr, idr,

```

```

kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
  izrez=izrez*izrez;

  izr=izdelm[k];
  kcr=kcdelm[k];
  kdr=kddelm[k];
  for(l=1;l<=kcr;l++)
  icr[l]=icdelm[(k-1)*k50+1];
  for(l=1;l<=kdr;l++)
  idr[l]=iddelm[(k-1)*k50+1];
  slzn(izr,izrez,&izr1,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
  izdelm[k]=izr1;
  kcdelm[k]=kcl;
  kddelm[k]=kdl;
  for(l=1;l<=kcl;l++)
  icdelm[(k-1)*k50+1]=icl[l];

  for(l=1;l<=kdl;l++)
  iddelm[(k-1)*k50+1]=idl[l];
//54 continue

}
//      printf("\n 54\n");
//c  pause '54'
//c  delm(k)=delm(k)/n
  izr=izdelm[k];
  kcr=kcdelm[k];
  kdr=kddelm[k];
  for(l=1;l<=kcr;l++)
  icr[l]=icdelm[(k-1)*k50+1];
  for(l=1;l<=kdr;l++)
  idr[l]=iddelm[(k-1)*k50+1];
  delzn(izr,izn,&izrez,kcr,kdr,icr,idr,
kcn,kdn,icn,idn,
&kcrez,&kdrez,icrez,idrez,krazr);
  izdelm[k]=izrez;
  kcdelm[k]=kcrez;
  kddelm[k]=kdrez;
  for(l=1;l<=kcrez;l++)
  icdelm[(k-1)*k50+1]=icrez[l];
  for(l=1;l<=kdrez;l++)
  iddelm[(k-1)*k50+1]=idrez[l];
  k=k+1;
  if(k<=kmax) goto m403;
//401 continue
m401:;
//      printf("\n m401\n");
//c  pause '401'
  if((igau==0)&&(k==1)) goto m9988;
  if(k<kmax) kmax=k-1;
  if((k==kmax)&&(igau==0)) kmax=k-1;
  izxmi=izdelm[kna];
  kcxmi=kcdelm[kna];
  kdxmi=kddelm[kna];
  for(l=1;l<=kcxmi;l++)
  icxmi[l]=icdelm[(kna-1)*k50+1];
  for(l=1;l<=kdxmi;l++)
  idxmi[l]=iddelm[(kna-1)*k50+1];
//c  xmin=delm(kna)
  imin=kna;

```

```

// do 405 i=kna,kmax
for(i=kna;i<=kmax;i++)
{
    kcel=kcdelm[i];
    kdel=kddelm[i];
    if(kcel>kcxmi) goto m405;
    if(kcel<kcxmi) goto m12101;
    for(l=1;l<=kcel;l++)
    icr[l]=icdelm[(i-1)*k50+1];
    for(l=1;l<=kdel;l++)
    idr[l]=iddelm[(i-1)*k50+1];
    for(l=1;l<=kcel;l++)
    {
        if(icr[l]>icxmi[l]) goto m405;
        if(icr[l]<icxmi[l]) goto m12101;
    }
    kdro=kdel;
    if(kdel>kdxmi) kdro=kdxmi;
    for(l=1;l<=kdro;l++)
    {
        if(idr[l]>idxmi[l]) goto m405;
        if(idr[l]<idxmi[l]) goto m12101;
    }

    if(kdel>=kdxmi) goto m405;

//c if(kdel.eq.kdxmi) goto 405

//c if(delm(i).ge.xmin) goto 405
m12101;;
//c xmin=delm(i)
    izxmi=izdelm[i];
    kcxmi=kcdelm[i];
    kdxmi=kddelm[i];
    for(l=1;l<=kcxmi;l++)
    icxmi[l]=icdelm[(i-1)*k50+1];

    for(l=1;l<=kdxmi;l++)
    idxmi[l]=iddelm[(i-1)*k50+1];
    imin=i;
m405;;
}

//с-проверить необходимость метки 406
//406 continue
//c pause '406'
//с-расчет модельных значений ум(i) по min ошибке
    fprintf(fp6, "\ноптимальная степень=%d\n", imin);

    fprintf(fp6, "\nМодель для зависимого параметра\n");
    kprod=7;
    iprod=0;
    ipech=0;
// do 4001 iarg=1,karg
for(iarg=1;iarg<=karg;iarg++)
{
    narg=mnarg[iarg];
    fprintf(fp6, "\наргумент-%d\n", narg);
    for(jl=1;jl<=60;jl++)

```

```

fprintf(fp6,"%c",anz[narg][j1]);

km1=imin*m+1;
km=imin*m;
kr=imin;

in=(iarg-1)*imin+1;
ik=in+imin-1;

// do 876 j=in,ik
for(j=in;j<=ik;j++)
{

// write(6,12451) narg,kr
fprintf(fp6,"\номер аргумента-%d степень=%d\n",narg,kr);
// k31=100;
iz1=izakm[(imin-1)*k31+j];
kcel=kcakm[(imin-1)*k31+j];
kdel=kdakm[(imin-1)*k31+j];
for(l=1;l<=kcel;l++)
icr[l]=icakm[((imin-1)*k31+j-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[((imin-1)*k31+j-1)*k50+1];

fprintf(fp6,"\знак=%d\n",iz1);

fprintf(fp6,"\целая часть коэффициента\n");
for(l=1;l<=kcel;l++)
fprintf(fp6,"%1d",icr[l]);
fprintf(fp6,"\n");
fprintf(fp6,"\дробная часть коэффициента\n");
for(l=1;l<=kdel;l++)
fprintf(fp6,"%1d",idr[l]);
fprintf(fp6,"\n");
//c write(9,50211) akm(imin,j),narg,kr
//c write(6,50211) akm(imin,j),narg,kr

kr=kr-1;
m876:;
}
m4001:;
}
//c write(6,14007) akm(imin,km1)
//c write(9,14007) akm(imin,km1)
//c write(*,14007) akm(imin,km1)
//c pause
iz=izakm[(imin-1)*k31+km1];
kcel=kcakm[(imin-1)*k31+km1];
kdel=kdakm[(imin-1)*k31+km1];
for(l=1;l<=kcel;l++)
icr[l]=icakm[((imin-1)*k31+km1-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[((imin-1)*k31+km1-1)*k50+1];

fprintf(fp6,"\свободный член модели\n");

fprintf(fp6,"\знак свободного члена=%d\n",iz);

fprintf(fp6,"\целая часть свободного члена\n");

```



```

icw[l]=1;
idw[l]=0;
izr=izx[(i-1)*m+j];
kcr=kcx[(i-1)*m+j];
kdr=kdx[(i-1)*m+j];
for(l=1;l<=kcr;l++)
icr[l]=icx[((i-1)*m+j-1)*k50+1];

for(l=1;l<=kdr;l++)
idr[l]=idx[((i-1)*m+j-1)*k50+1];

if(istep==0) goto m31103;
// do 31104 isw=1,istep
for(isw=1;isw<=istep;isw++)
{
ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
izw=izrez;
kcw=kcrez;
kdw=kdrez;
for(l=1;l<=kcw;l++)
icw[l]=icrez[l];
for(l=1;l<=kdw;l++)
idw[l]=idrez[l];

//31104 continue
}
m31103:;
izr=izakm[(imin-1)*k31+ist];
kcr=kcakm[(imin-1)*k31+ist];
kdr=kdakm[(imin-1)*k31+ist];
for(l=1;l<=kcr;l++)
icr[l]=icakm[((imin-1)*k31+ist-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idakm[((imin-1)*k31+ist-1)*k50+1];

ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
iz1=izym[i];
kcr=kcym[i];
kdr=kdym[i];
for(l=1;l<=kcr;l++)
icr[l]=icym[(i-1)*k50+1];

for(l=1;l<=kdr;l++)
idr[l]=idym[(i-1)*k50+1];

slzn(iz1,izrez,&iz2,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
izym[i]=iz2;
kcym[i]=kcl;
kdym[i]=kdl;
for(l=1;l<=kcl;l++)
icym[(i-1)*k50+1]=icl[l];
for(l=1;l<=kdl;l++)
idym[(i-1)*k50+1]=idl[l];

//c ym(i)=ym(i)+akm(imin,ist)*x(i,j)**(imin-(isp-1))

iz1=izwkl[j];

```

```

kcel=kcwkl[j];
kdel=kdwkl[j];
for(l=1;l<=kcel;l++)
icr[l]=icwkl[(j-1)*k50+1];
for(l=1;l<=kdel;l++)
idr[l]=idwkl[(j-1)*k50+1];
izrez=izrez*izrez;
slzn(iz1,izrez,&iz2,kcel,kdel,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
//c   wk1(j)=wk1(j)+dabs(akm(imin,ist)*x(i,j)**(imin-(isp-1)))

izwkl[j]=iz2;
kcwkl[j]=kcl;
kdwkl[j]=kdl;
for(l=1;l<=kcl;l++)
icwkl[(j-1)*k50+1]=icl[l];
for(l=1;l<=kdl;l++)
idwkl[(j-1)*k50+1]=idl[l];

isp=isp+1;
//3355   continue
}
//33540   continue
}
izr=izym[i];
kcr=kcym[i];
kdr=kdym[i];
for(l=1;l<=kcr;l++)
icr[l]=icym[(i-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idym[(i-1)*k50+1];

iz1=izakm[(imin-1)*k31+km1];
kcl=kcakm[(imin-1)*k31+km1];
kdl=kdakm[(imin-1)*k31+km1];
for(l=1;l<=kcl;l++)
icl[l]=icakm[((imin-1)*k31+km1-1)*k50+1];
for(l=1;l<=kdl;l++)
idl[l]=idakm[((imin-1)*k31+km1-1)*k50+1];
wiczn(izr,iz1,&izrez,kcr,kdr,icr,idr,
kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
//c   sy=sy+ym(i)-akm(imin,km1)
      slzn(izsy,izrez,&iz2,kcsy,kdsy,icsy,idsy,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
izsy=iz2;
kcsy=kcl;
kdsy=kdl;
for(l=1;l<=kcl;l++)
icsy[l]=icl[l];
for(l=1;l<=kdsy;l++)
idsy[l]=idl[l];
izr=izym[i];
kcr=kcym[i];
kdr=kdym[i];
for(l=1;l<=kcr;l++)
icr[l]=icym[(i-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idym[(i-1)*k50+1];
/*c   write(6,*) i
c     write(6,*) 'модельное y'

```

```

c      write(6,*) izr
c      write(6,*) 'целая часть'
c      write(6,87321) (icr(l),l=1,kcr)
87321 format(1x,60i1)
c      write(6,*) 'дробная часть'
c      write(6,87321) (idr(l),l=1,kdr) */
      iz1=izy[i];
      kcl=кcy[i];
      kdl=кdy[i];
      for(l=1;l<=kcl;l++)
      icl[l]=icy[(i-1)*k50+1];
      for(l=1;l<=kdl;l++)
      idl[l]=idy[(i-1)*k50+1];
/*c    write(6,*) 'исходное y'

c      write(6,*) iz1

c      write(6,*) 'целая часть'
c      write(6,87321) (icl(l),l=1,kcl)
c      write(6,*) 'дробная часть'
c      write(6,87321) (idl(l),l=1,kdl)      */
      wiczn(izr,iz1,&iz2,kcr,kdr,icr,idr,
      kcl,kdl,icl,idl,
      &kcrez,&kdrez,icrez,idrez,krazr);
      izrez=iz2*iz2;
//c    delt=delt+dabs(ym(i)-y(i))
      slzn(izdelt,izrez,&iz2,kcdelt,kddelt,icdelt,iddelt,
      kcrez,kdrez,icrez,idrez,
      &kcl,&kdl,icl,idl,krazr);
      izdelt=iz2;
      kcdelt=kcl;
      kddelt=kdl;
      for(l=1;l<=kcl;l++)

      icdelt[l]=icl[l];
      for(l=1;l<=kdl;l++)
      iddelt[l]=idl[l];

//c345    format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
m3354:;
      }
      izwtot=1;
      kcwtot=1;
      kdwtot=1;
      icwtot[1]=0;
      idwtot[1]=0;
//c    wkltot=0
//    do 5204 i=1,m
      for(i=1;i<=m;i++)
      {
      izr=1;
      kcr=kcwkl[i];
      kdr=kdwkl[i];
      for(l=1;l<=kcr;l++)
      icr[l]=icwkl[(i-1)*k50+1];
      for(l=1;l<=kdr;l++)
      idr[l]=idwkl[(i-1)*k50+1];
//c    wkltot=wkltot+dabs(wkl(i))
      slzn(izwtot,izr,&iz1,kcwtot,kdwtot,icwtot,idwtot,
      kcr,kdr,icr,idr,
      &kcrez,&kdrez,icrez,idrez,krazr);
      izwtot=iz1;
      kcwtot=kcrez;

```

```

        kdwtot=kdrez;
        for (l=1;l<=kcrez;l++)
        icwtot[l]=icrez[l];
        for (l=1;l<=kdrez;l++)
        idwtot[l]=idrez[l];
//5204        continue
    }
    for (i=1;i<=m;i++)
    {
        izr=izwkl[i];
        kcr=kcwkl[i];
        kdr=kdwkl[i];
        for (l=1;l<=kcr;l++)
        icr[l]=icwkl[(i-1)*k50+1];
        for (l=1;l<=kdr;l++)
        idr[l]=idwkl[(i-1)*k50+1];
//c        wk1(i)=wk1(i)/wkltot
        delzn(izr,izwtot,&izl,kcr,kdr,icr,idr,
        kcwtot,kdwtot,icwtot,idwtot,
        &kc1,&kdl,ic1,id1,krazr);
        izwkl[i]=izl;
        kcwkl[i]=kc1;
        kdwkl[i]=kdl;
        for (l=1;l<=kc1;l++)
            icwkl[(i-1)*k50+1]=ic1[l];
        for (l=1;l<=kdl;l++)
            idwkl[(i-1)*k50+1]=id1[l];
//5203        continue
//5202        continue
    }
//c    delt=delt/n

    delzn(izdelt,izn,&izr,kcdelt,kddelt,icdelt,iddelt,
    kcn,kdn,icn,idn,
    &kcrez,&kdrez,icrez,idrez,krazr);
    izdelt=izr;
    kcdelt=kcrez;
    kddelt=kdrez;
    for (l=1;l<=kcdelt;l++)
    icdelt[l]=icrez[l];
    for (l=1;l<=kddelt;l++)
    iddelt[l]=idrez[l];

    fprintf(fp6,"\nсредняя абсолютная ошибка\n");
    fprintf(fp6,"\nцелая часть ошибки\n");
    for (l=1;l<=kcdelt;l++)
    fprintf(fp6,"%1d",icdelt[l]);
    fprintf(fp6,"\nдробная часть ошибки\n");
    for (l=1;l<=kddelt;l++)
    fprintf(fp6,"%1d",iddelt[l]);

    fprintf(fp6,"\n");
//    prob=' '
//    do 5404 i=1,m
    for (i=1;i<=m;i++)
    {
        narg=mnarg[i];
//    write(6,5405) narg,(anaz(narg,j),j=1,60)
        fprintf(fp6,"\nномер аргумента=%d\n",narg);
        for (j=1;j<=60;j++)

```

```
fprintf(fp6,"%c",anaz[narg][j]);
    fprintf(fp6,"\nцелая часть вклада\n");
kcel=kcwkl[i];
for(l=1;l<=kcel;l++)
    fprintf(fp6,"%1d",icwkl[(i-1)*k50+l]);
fprintf(fp6,"\ndробная часть вклада\n");
kdel=kdwkl[i];
for(l=1;l<=kdel;l++)
    fprintf(fp6,"%1d",idwkl[(i-1)*k50+l]);
    fprintf(fp6,"\n");

//5404     continue
    }
//891 format(/1x,'средняя абсолютная ошибка=',f36.18/)
m9988;;
m2001;;
    }

fclose(fp4);
fclose(fp6);
fclose(fp5);
fclose(fp7);
return;
}
```

## Набор mnk.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

void mnk(int n,int m,int izn,int kcn,
int kdn,int *icn,int *idn,int *izx,int *kcx,int *kdx,
int *icx,int *idx,
int *izy,int *kcy,int *kdy,
int *icy,int *idy,int k,int *izak,int *kcak,int *kdak,
int *icak,int *idak,int *pigau,int krazr,int k50)
{
extern FILE *fp6,*fp9;
extern void ymnzn(int izw,int izr,int *pizrez,
int kcw,int kdw,int *icw,int *idw,
int kcr,int kdr,int *icr,int *idr,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern void slzn(int izr,int izrez,int*pizl,
int kcr,int kdr,int *icr,int *idr,
int kcrez,int kdrez,int *icrez,int *idrez,
int *pkcl,int *pkdl,int *icl,int *idl,int krazr);

extern void gaydl(int kg,int *izc,int *kcc,int *kdc,
int *icc,int *idc,
int *izf,int *kcf,int *kdf,int *icf,int *idf,
int *izak,int *kcak,int *kdak,int *icak,
int *idak,int *pigau,int krazr,int k50,int *mnper);

extern void prowg(int kg,int *izc,int *kcc,int *kdc,
int *icc,int *idc,
int *izf,int *kcf,int *kdf,int *icf,int *idf,
int *izak,int *kcak,int *kdak,int *icak,int *idak,
int *pigau,int krazr,int k50);

//с-метод наименьших квадратов в длинной арифметике
int *izf,*kcf,*kdf,*icf,*idf,mnper[100],
*pizrez,*pkcrez,*pizl,*pkcl,*pkdl,
icw[400],idw[400],ilf,
//izx(50,10),kcx(50,10),kdx(50,10)
// integer*2 icx(50,10,20),idx(50,10,20),
icrez[400],idrez[400],
// *izy(50),kcy(50),kdy(50),icy(50,20),idy(50,20),
icr[400],idr[400],
*izc,*kcc,*kdc,*icc,*idc,
icwl[400],idwl[400],icl[400],idl[400];
//icn(400),idn(400)
// integer*2 izak(31),kcak(31),kdak(31),icak(31,20),idak(31,20)
int s,p,jp,ip,igau,igur;
int k1,i,izw,kcw,kdw,izr,kcr,kdr,l,istep,izrez,kcrez,kdrez,lp,izl,
kcl,kdl,j,jlc,ilc,kzmat,lpr,izwl,kdw1,kcw1,kmat,istr,isto,inf,
iccl,jcl,kraz,kcel,kdel,kg;
kzmat=k*m+1;kmat=k*m+1;
izf=(int *)farmalloc(101*12);
if(izf==NULL)

```

```

{
fprintf(fp6,"NO MEMORY FOR izf \n");
return;
}
kcf =(int *)farmalloc(101*12);
if(kcf ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcf \n");
return;
}
kdf =(int *)farmalloc(101*12);
if(kdf ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdf \n");
return;
}
icf =(int *)farmalloc(2200*12);
if(icf ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icf \n");
return;
}
idf =(int *)farmalloc(2200*12);
if(idf ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idf \n");
return;
}
izc =(int *)farmalloc(10300*12);
if(izc ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izc \n");
return;
}
kcc =(int *)farmalloc(10300*12);
if(kcc ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcc \n");
return;
}
kdc =(int *)farmalloc(10300*12);
if(kdc ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdc \n");
return;
}
icc=(int *)farmalloc(215000*12);
if(icc ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icc \n");
return;
}
idc =(int *)farmalloc(215000*12);
if(idc ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idc \n");
return;
}
}
//
//с-с(50,50), матрица системы нормальных уравнений
//с-f(50), вектор свободных членов
//с-ak(50), коэффициенты полиномов по аргументам

```

```

//с-х(50,10), матрица аргументов
//с-у(50), массив значений функции
//с-ум(50), массив модельных значений функции
//с write(6,*) ' Метод наименьших квадратов '
/* с-определение матрицы с
с-і-номер наблюдения, строки
с-j-номер параметра-аргумента
с-l-номер коэффициента, степени
с-s-номер степени, р-номер параметра
с-m-количество параметров-аргументов
с-k, порядок полиномов
с-n, число значений функции */
// fprintf(fp6, "\nпередача в mnk\n");
// fprintf(fp6, "n=%d m=%d k50=%d krazr=%d k=%d\n", n, m,
// k50, krazr, k);
// printf("\n mnk n=%d m=%d k50=%d krazr=%d\n", n, m, k50, krazr);
// scanf("%d", &iz1);
// fprintf(fp6, "\nматрица x\n");
//c do 34521 i=1, n
// for(i=1; i<=n; i++)
// {
//c do 34522 j=1, m
// for(j=1; j<=m; j++)
// {
// fprintf(fp6, "\n i=%d j=%d\n", i, j);
// fprintf(fp6, "\nцелая часть x\n");
// iz1=izx[(i-1)*m+j];
// fprintf(fp6, "\nзнак=%d \n", iz1);
// kcel=kcx[(i-1)*m+j];
// kdel=kdx[(i-1)*m+j];
// for(l=1; l<=kcel; l++)
// fprintf(fp6, "%d", icx[((i-1)*m+j-1)*k50+1]);

// fprintf(fp6, "\ndробная часть x\n");
// for(l=1; l<=kdel; l++)
// fprintf(fp6, "%d", idx[((i-1)*m+j-1)*k50+1]);
//34522 continue
//34521 continue
// }
// }
// fprintf(fp6, "\nвектор y\n");
// do 34545 i=1, n
/* for(i=1; i<=n; i++)
{
kcel=ky[i];
kdel=ky[i];
fprintf(fp6, "\n mnk знак=%d kcel=%d kdel=%d \n",
izy[i], kcel, kdel);
// printf(" \nmnk zn=%d kcel=%d kdel=%d\n", izy[i], kcel, kdel);
// scanf("%d", &iz1);
// fprintf(fp6, "\nцелая часть y\n");
// printf("\nцелая часть\n");
for(l=1; l<=kcel; l++)
{
// printf("%d", icy[(i-1)*k50+1]);
// scanf("%d", &iz1);
// fprintf(fp6, "%d", icy[(i-1)*k50+1]);
}
// printf("\ndробная часть\n");
// fprintf(fp6, "\ndробная часть y \n");
for(l=1; l<=kdel; l++)
{
// printf("%d", idy[(i-1)*k50+1]);

```

```

//  scanf("%d",&iz1);
//  fprintf(fp6,"%d",idy[(i-1)*k50+1]);
//  }
//  printf("\ni=%d\n",i);
//  scanf("%d",&iz1);
//34545  continue
//  }
//c  stop  */
//  k1=k+1;
//c  pause 'mnk1'
//  do 2 p=1,m
//  printf(" mnk do2 \n");
//  scanf("%d",&igur);
//  return;
//  fprintf(fp6,"\nwid y\n");
//  for(p=1;p<=m;p++)
//  {

//  do 282 s=1,k
//  for(s=1;s<=k;s++)
//  {
//  fprintf(fp6,"\n p=%d s=%d\n",p,s);
//  izf[(p-1)*k+s]=1;
//  kcf[(p-1)*k+s]=1;
//  kdf[(p-1)*k+s]=1;
//  icf[((p-1)*k+s-1)*k50+1]=0;
//  idf[((p-1)*k+s-1)*k50+1]=0;
//c  f((p-1)*k+s)=0
//  do 2002 i=1,n
//  for(i=1;i<=n;i++)
//  {
//  fprintf(fp6,"\n i=%d\n",i);
//  izw=1;
//  kcw=1;
//  kdw=1;
//  icw[1]=1;
//  idw[1]=0;
//  izr=izx[(i-1)*m+p];
//  kcr=kcx[(i-1)*m+p];
//  kdr=kdx[(i-1)*m+p];
//  for(l=1;l<=kcr;l++)
//  icr[l]=icx[((i-1)*m+p-1)*k50+1];
//  for(l=1;l<=kdr;l++)
//  idr[l]=idx[((i-1)*m+p-1)*k50+1];
//  istep=k-(s-1);
//  if(istep==0) goto m32103;
//  do 12104 l=1,istep
//  for(l=1;l<=istep;l++)
//  {

//  ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
//  kcr,kdr,icr,idr,
//  &kcrez,&kdrez,icrez,idrez,krazr);
//c  write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez
//c  pause '2'
//  izw=izrez;
//  kcw=kcrez;
//  kdw=kdrez;
//  for(lp=1;lp<=kcrez;lp++)
//  icw[lp]=icrez[lp];
//  for(lp=1;lp<=kdrez;lp++)
//  idw[lp]=idrez[lp];
//12104  continue

```

```

    }

m32103:;
//   fprintf(fp6, "\nm32103\n");
//c   pause '32103'
      izr=izy[i];
      kcr=kcy[i];
      kdr=kdy[i];
      for(l=1;l<=kcr;l++)
        icr[l]=icy[(i-1)*k50+l];
      for(l=1;l<=kdr;l++)
        idr[l]=idy[(i-1)*k50+l];
      ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
//   fprintf(fp6, "\nposle ymnzn\n");
      ilf=(p-1)*k+s;
      izr=izf[ilf];
      kcr=kcf[ilf];
      kdr=kdf[ilf];
      for(l=1;l<=kcr;l++)
        icr[l]=icf[(ilf-1)*k50+l];
      for(l=1;l<=kdr;l++)
        idr[l]=idf[(ilf-1)*k50+l];
/* c   pause '5 na'
c     write(6,*) 'первое слагаемое'
c     write(6,*) izr,kcr,kdr
c     write(6,*) izr,kcr,kdr
c     pause '5_1'
c     write(6,*) izr

c     write(6,49701) izr
49701 format(1x,'знак=',i3/)
c     write(6,*) 'целая часть'
c     write(6,*) (icr(l),l=1,kcr)
c     pause '5_1_1'
c     write(6,*) (icr(l),l=1,kcr)
c     pause
49702 format(1x,50i1)
c     write(6,*) 'дробная часть'
c     write(6,*) (idr(l),l=1,kdr)
49703 format(1x,50i1)
c     write(6,*) 'второе слагаемое'
c     write(6,49701) izrez
c     write(6,*) izrez,kcrez,kdrez
c     pause '5_2'
c     write(6,*) 'целая часть'
c     write(6,*) (icrez(l),l=1,kcrez)
c     write(6,*) 'дробная часть'
c     write(6,*) (idrez(l),l=1,kdrez)
c     pause '5' */
      slzn(izr,izrez,&izl,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
//   fprintf(fp6, "\nposle slzn\n");
//c   pause '6'
//c   f((p-1)*k+s)= f((p-1)*k+s)+y(i)*x(i,p)**(k-(s-1))
      izf[ilf]=izl;
      kcf[ilf]=kcl;
      kdf[ilf]=kdl;
      for(l=1;l<=kcl;l++)
        icf[(ilf-1)*k50+l]=icl[l];
      for(l=1;l<=kdl;l++)

```

```

        idf[(ilf-1)*k50+1]=idl[1];
//      fprintf(fp6,"\npered 2002\n");
//2002      continue
    }
//      fprintf(fp6,"\n 2002\n");
//      scanf("%d",&igur);
//c      pause '2002'
//      do 212 j=1,m
//          for(j=1;j<=m;j++)
//              {

//          do 211 l=1,k
//              for(l=1;l<=k;l++)
//                  {
//          fprintf(fp6," \nmnk j=%d l=%d\n",j,l);
//          ilc=(p-1)*k+s;
//          jlc=(j-1)*k+1;
//c          c((p-1)*k+s,(j-1)*k+1)=0
//          kzmat=k*m+1;

//          izc[(ilc-1)*kzmat+jlc]=1;
//          kcc[(ilc-1)*kzmat+jlc]=1;
//          kdc[(ilc-1)*kzmat+jlc]=1;
//          icc[((ilc-1)*kzmat+jlc-1)*k50+1]=0;
//          idc[((ilc-1)*kzmat+jlc-1)*k50+1]=0;
//          do 3 i=1,n
//              for(i=1;i<=n;i++)
//                  {

//c          c((p-1)*k+s,(j-1)*k+1)= c((p-1)*k+s,(j-1)*k+1)+
//c          *x(i,j)**(k-(l-1))*x(i,p)**(k-(s-1))
//          izw=1;
//          kcw=1;
//          kdw=1;
//          icw[1]=1;
//          idw[1]=0;
//          istep=k-(l-1);
//          izr=izx[(i-1)*m+j];
//          kcr=kcx[(i-1)*m+j];
//          kdr=kdx[(i-1)*m+j];
//          for(lp=1;lp<=kcr;lp++)
//              icr[lp]=icx[((i-1)*m+j-1)*k50+lp];
//          for(lp=1;lp<=kdr;lp++)
//              idr[lp]=idx[((i-1)*m+j-1)*k50+lp];
//          if(istep==0) goto m16101;
//          do 16102 lpr=1,istep
//              for(lpr=1;lpr<=istep;lpr++)
//                  {

//c          pause '7'
//          ymnzn(izw,izr,&izl,kcw,kdw,icw,idw,
//          kcr,kdr,icr,idr,
//          &kcrez,&kdrez,icrez,idrez,krazr);
//c          pause '8'
//          izw=izl;
//          kcw=kcrez;
//          kdw=kdrez;
//          for(lp=1;lp<=kcw;lp++)
//              icw[lp]=icrez[lp];
//          for(lp=1;lp<=kdw;lp++)
//              idw[lp]=idrez[lp];
//16102      continue
//          }

```

```

m16101:;
//   fprintf(fp6,"\n m1601\n");
//   scanf("%d",&igur);
//   istep=k-(s-1);
//   izw1=1;
//   kcwl=1;
//   kdwl=1;
//   icwl[1]=1;
//   idwl[1]=0;
//   izr=izx[(i-1)*m+p];
//   kcr=kcx[(i-1)*m+p];
//   kdr=kdx[(i-1)*m+p];
//   for(lp=1;lp<=kcr;lp++)
//   icr[lp]=icx[((i-1)*m+p-1)*k50+lp];
//   for(lp=1;lp<=kdr;lp++)
//   idr[lp]=idx[((i-1)*m+p-1)*k50+lp];
//   if(istep==0) goto m18301;
//   do 18302 lpr=1,istep
//     for(lpr=1;lpr<=istep;lpr++)
//       {

//c   pause '9'
//     ymnzn(izw1,izr,&izrez,kcwl,kdwl,icwl,idwl,
//     kcr,kdr,icr,idr,
//     &kcrez,&kdrez,icrez,idrez,krazr);
//c   pause '10'
//     izw1=izrez;
//     kcwl=kcrez;
//     kdwl=kdrez;
//     for(lp=1;lp<=kcrez;lp++)
//     icwl[lp]=icrez[lp];
//     for(lp=1;lp<=kdrez;lp++)
//     idwl[lp]=idrez[lp];

//18302   continue
//       }

m18301:;
//   fprintf(fp6,"\nm18301\n");
//   scanf("%d",&igur);
//c   pause '11'
//   fprintf(fp6,"\n per ymnzn...\n");
//   ymnzn(izw,izw1,&izrez,kcw,kdw,icw,idw,
//   kcwl,kdwl,icwl,idwl,
//   &kcrez,&kdrez,icrez,idrez,krazr);
//   fprintf(fp6,"\n 12\n");
//c   pause '12'
//   kmat=k*m+1;
//   fprintf(fp6,"\nmnk kmat=%d\n",kmat);
//   scanf("%d",&igur);
//   fprintf(fp6,"\nizc[]=%d kcc[]=%d kdc[]=%d\n",
//   izc[(ilc-1)*kmat+jlc],kcc[(ilc-1)*kmat+jlc],
//   kdc[(ilc-1)*kmat+jlc]);
//   fprintf(fp6,"\n ilc=%d jlc=%d\n",ilc,jlc);
//   izr=izc[(ilc-1)*kmat+jlc];
//   kcr=kcc[(ilc-1)*kmat+jlc];
//   kdr=kdc[(ilc-1)*kmat+jlc];
//   fprintf(fp6,"\n izr=%d kcr=%d kdr=%d\n",izr,kcr,kdr);
//   for(lp=1;lp<=kcr;lp++)
//   icr[lp]=icc[((ilc-1)*kmat+jlc-1)*k50+lp];
//   for(lp=1;lp<=kdr;lp++)
//   idr[lp]=idc[((ilc-1)*kmat+jlc-1)*k50+lp];
//   fprintf(fp6,"\n 13\n");

```

```

//  scanf("%d",&igur);
//c  pause '13'
//  fprintf(fp6,"\nper sl....\n");
    slzn(izrez,izr,&izl,kcrez,kdrez,icrez,idrez,
    kcr,kdr,icr,idr,

    &kcl,&kdl,icl,idl,krazr);
//  fprintf(fp6,"\n 14\n");
//  scanf("%d",&igur);
//c  pause '14'
    izc[(ilc-1)*kmat+jlc]=izl;
    kcc[(ilc-1)*kmat+jlc]=kcl;
    kdc[(ilc-1)*kmat+jlc]=kdl;
    for(lp=1;lp<=kcl;lp++)
    icc[((ilc-1)*kmat+jlc-1)*k50+lp]=ic1[lp];
    for(lp=1;lp<=kdl;lp++)
    idc[((ilc-1)*kmat+jlc-1)*k50+lp]=id1[lp];
//3  continue
    }
//  fprintf(fp6,"\n 3\n");
//  scanf("%d",&igur);
    istr=(p-1)*k+s;
    isto=(j-1)*k+l;
//211  continue
    }
//  fprintf(fp6,"\n 211\n");
//212  continue
    }
//  fprintf(fp6,"\n212\n");
//  scanf("%d",&igur);
    ilc=(p-1)*k+s;
    jlc=(m-1)*k+k+1;
    izc[(ilc-1)*kmat+jlc]=1;
    kcc[(ilc-1)*kmat+jlc]=1;
    kdc[(ilc-1)*kmat+jlc]=1;
    icc[((ilc-1)*kmat+jlc-1)*k50+1]=0;
    idc[((ilc-1)*kmat+jlc-1)*k50+1]=0;
//c  c((p-1)*k+s,(m-1)*k+k+1)=0
//  do 703 i=1,n
    for(i=1;i<=n;i++)
    {
//  fprintf(fp6,"\n 703 i=%d\n",i);
//  scanf("%d",&igur);
    istep=k-(s-1);
    izw=1;
    kcw=1;
    kdw=1;
    icw[1]=1;
    idw[1]=0;
    izr=izx[(i-1)*m+p];
    kcr=kcx[(i-1)*m+p];
    kdr=kdx[(i-1)*m+p];
    for(lp=1;lp<=kcr;lp++)
    icr[lp]=icx[((i-1)*m+p-1)*k50+lp];
    for(lp=1;lp<=kdr;lp++)
    idr[lp]=idx[((i-1)*m+p-1)*k50+lp];
    if(istep==0) goto m18403;
//  do 18404 lpr=1,istep
    for(lpr=1;lpr<=istep;lpr++)
    {
//c  pause '15'
    ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
    kcr,kdr,icr,idr,

```

```

&kcrez, &kdrez, icrez, idrez, krazr);
//c  pause '16'
    izw=izrez;
    kcw=kcrez;
    kdw=kdrez;
    for (lp=1; lp<=kcrez; lp++)

        icw[lp]=icrez[lp];
        for (lp=1; lp<=kdrez; lp++)
            idw[lp]=idrez[lp];
//18404  continue
    }

m18403:;
//  fprintf(fp6, "\n m18403\n");
//  scanf("%d", &igur);
    izr=izc[(ilc-1)*kmat+jlc];
    kcr=kcc[(ilc-1)*kmat+jlc];
    kdr=kdc[(ilc-1)*kmat+jlc];
    for (lp=1; lp<=kcr; lp++)
        icr[lp]=icc[((ilc-1)*kmat+jlc-1)*k50+lp];
    for (lp=1; lp<=kdr; lp++)
        idr[lp]=idc[((ilc-1)*kmat+jlc-1)*k50+lp];
//  printf("\npered slzn\n");
//  scanf("%d", &igur);
    slzn(izw, izr, &izrez, kcw, kdw, icw, idw,
        kcr, kdr, icr, idr,
        &kcrez, &kdrez, icrez, idrez, krazr);
//  printf("\nposle slzn\n");
//  scanf("%d", &igur);
    izc[(ilc-1)*kmat+jlc]=izrez;
    kcc[(ilc-1)*kmat+jlc]=kcrez;
    kdc[(ilc-1)*kmat+jlc]=kdrez;
    for (lp=1; lp<=kcrez; lp++)
        icc[((ilc-1)*kmat+jlc-1)*k50+lp]=icrez[lp];
    for (lp=1; lp<=kdrez; lp++)
        idc[((ilc-1)*kmat+jlc-1)*k50+lp]=idrez[lp];
//c  c((p-1)*k+s, (m-1)*k+k+1) = c((p-1)*k+s, (m-1)*k+k+1) +
//c  *x(i, p)**(k-(s-1))
//  printf("\npered 703\n");
//  scanf("%d", &igur);
//703  continue
    }
//  fprintf(fp6, "\nposle 703\n");
//  scanf("%d", &igur);
    istr=(p-1)*k+s;
    isto=(m-1)*k+k+1;
//c  write(6, 3311) p, s, j, l, istr, isto, c(istr, isto), f(istr)
//282 continue
    }
//2  continue
    }
//  fprintf(fp6, "\n posle 2\n");
//  scanf("%d", &igur);
    inf=(m-1)*k+k+1;
    izf[inf]=1;
    kcf[inf]=1;
    kdf[inf]=1;
    icf[(inf-1)*k50+1]=0;
    idf[(inf-1)*k50+1]=0;
//c  f((m-1)*k+k+1)=0
//  do 12002 i=1, n
    for (i=1; i<=n; i++)

```

```

{
  izr=izf[inf];
  kcr=kcf[inf];
  kdr=kdf[inf];
  for(lp=1;lp<=kcr;lp++)
  icr[lp]=icf[(inf-1)*k50+lp];
  for(lp=1;lp<=kdr;lp++)
  idr[lp]=idf[(inf-1)*k50+lp];
  izl=izy[i];
  kcl=kcy[i];
  kdl=kdy[i];
  for(lp=1;lp<=kcl;lp++)
  icl[lp]=icy[(i-1)*k50+lp];
  for(lp=1;lp<=kdl;lp++)
  idl[lp]=idy[(i-1)*k50+lp];

  slzn(izr,izl,&izrez,kcr,kdr,icr,idr,
kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
  izf[inf]=izrez;
  kcf[inf]=kcrez;
  kdf[inf]=kdrez;
  for(lp=1;lp<=kcrez;lp++)
  icf[(inf-1)*k50+lp]=icrez[lp];
  for(lp=1;lp<=kdrez;lp++)
  idf[(inf-1)*k50+lp]=idrez[lp];

//c f((m-1)*k+k+1)=f((m-1)*k+k+1)+y(i)
//12002 continue
}
// fprintf(fp6,"\npered 4212\n");
// do 4212 j=1,m
// for(j=1;j<=m;j++)
// {
// do 4211 l=1,k
// for(l=1;l<=k;l++)
// {
// fprintf(fp6,"\n 4211 j=%d l=%d\n",j,l);
  icc1=(m-1)*k+k+1;
  jc1=(j-1)*k+1;
  izc[(icc1-1)*kmat+jc1]=1;
  kcc[(icc1-1)*kmat+jc1]=1;
  kdc[(icc1-1)*kmat+jc1]=1;
  icc[(icc1-1)*kmat+jc1-1]*k50+1]=0;
  idc[(icc1-1)*kmat+jc1-1]*k50+1]=0;
//c c((m-1)*k+k+1,(j-1)*k+1)=0
// do 371 i=1,n
// for(i=1;i<=n;i++)
// {
// fprintf(fp6,"\n 371 i=%d\n",i);
  istep=k-(l-1);
  izw=1;
  kcw=1;
  kdw=1;
  icw[1]=1;
  idw[1]=0;
  izr=izx[(i-1)*m+j];
  kcr=kcx[(i-1)*m+j];
  kdr=kdx[(i-1)*m+j];
// fprintf(fp6,"\n371 izr=%d kcr=%d kdr=%d\n",
// izr,kcr,kdr);
  for(lp=1;lp<=kcr;lp++)
  icr[lp]=icx[((i-1)*m+j-1)*k50+lp];

```

```

        for(lp=1;lp<=kdr;lp++)
        idr[lp]=idx[((i-1)*m+j-1)*k50+lp];
//      fprintf(fp6,"\n 371 istep=%d\n",istep);
        if(istep==0) goto m23673;
//      do 23674 lpr=1,istep
        for(lpr=1;lpr<=istep;lpr++)
        {
//      fprintf(fp6,"\n 371 lpr=%d izr=%d kcr=%d kdr=%d\n",
//      lpr,izr,kcr,kdr);
//c      pause '21'
        ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
        kcr,kdr,icr,idr,
        &kcrez,&kdrez,icrez,idrez,krazr);
//      fprintf(fp6,"\n 371 posle ymnzn\n");
//      if(l==2) return;
        izw=izrez;
        kw=kcrez;
        kdw=kdrez;
//fprintf(fp6,"\n 371 izw=%d kw=%d kdw=%d\n",izw,kw,kdw);
//      if(l==2) return;
        for(lp=1;lp<=kcrez;lp++)
        icw[lp]=icrez[lp];
        for(lp=1;lp<=kdrez;lp++)
        idw[lp]=idrez[lp];

//23674      continue
    }
m23673:;
//      fprintf(fp6,"\nm23673 \n");
//      if(l==2) return;
        izr=izc[(iccl-1)*kmat+jc1];
        kcr=kcc[(iccl-1)*kmat+jc1];
        kdr=kdc[(iccl-1)*kmat+jc1];
        for(lp=1;lp<=kcr;lp++)
        icr[lp]=icc[((iccl-1)*kmat+jc1-1)*k50+lp];
        for(lp=1;lp<=kdr;lp++)
        idr[lp]=idc[((iccl-1)*kmat+jc1-1)*k50+lp];
        slzn(izw,izr,&izrez,kcw,kdw,icw,idw,
        kcr,kdr,icr,idr,
        &kcrez,&kdrez,icrez,idrez,krazr);

        izc[(iccl-1)*kmat+jc1]=izrez;
        kcc[(iccl-1)*kmat+jc1]=kcrez;
        kdc[(iccl-1)*kmat+jc1]=kdrez;
        for(lp=1;lp<=kcrez;lp++)
        icc[((iccl-1)*kmat+jc1-1)*k50+lp]=icrez[lp];
        for(lp=1;lp<=kdrez;lp++)
        idc[((iccl-1)*kmat+jc1-1)*k50+lp]=idrez[lp];
//c      c((m-1)*k+k+1,(j-1)*k+1)= c((m-1)*k+k+1,(j-1)*k+1)+
//c          *x(i,j)**(k-(l-1))
//371      continue
    }
//      fprintf(fp6,"\n 371\n");
//      if(l==2) return;
        istr=(m-1)*k+k+1;
        isto=(j-1)*k+1;
//4211      continue
    }
//4212      continue
    }
//      fprintf(fp6,"\n4212 \n");
//      if(k==2) return;
        iccl=(m-1)*k+k+1;

```

```

    ize[(iccl-1)*kmat+iccl]=izn;
    kcc[(iccl-1)*kmat+iccl]=kcn;
    kdc[(iccl-1)*kmat+iccl]=kdn;
    for(lp=1;lp<=kcn;lp++)
    icc[(iccl-1)*kmat+iccl-1]*k50+lp]=icn[lp];
    for(lp=1;lp<=kdn;lp++)
    idc[(iccl-1)*kmat+iccl-1]*k50+lp]=idn[lp];

//c   c((m-1)*k+k+1,(m-1)*k+k+1)=n

    istr=(m-1)*k+k+1;
    isto=(m-1)*k+k+1;
//c   write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)

    kraz=m*k+1;
//   fprintf(fp6,"\n zap c\n");
//   if(k==2) return;
//c   write(5,*) ' матрица c'
//   open(9,file='promprow',status='Unknown')
    fp9=fopen("promprc","w");
    fprintf(fp9,"%d\n",kraz);
//   do 56 i=1,kraz
    for(i=1;i<=kraz;i++)
    {
//   do 1234 j=1,kraz
    for(j=1;j<=kraz;j++)
    {
        izr=ize[(i-1)*kmat+j];
        kcel=kcc[(i-1)*kmat+j];
        kdel=kdc[(i-1)*kmat+j];

        fprintf(fp9,"%d %d %d \n",izr,kcel,kdel);
        for(jp=1;jp<=kcel;jp++)
            fprintf(fp9,"%d ",icc[(i-1)*kmat+j-1]*k50+jp]);
        fprintf(fp9,"\n");

        for(jp=1;jp<=kdel;jp++)
            fprintf(fp9,"%d ",idc[(i-1)*kmat+j-1]*k50+jp]);
        fprintf(fp9,"\n");
//1234        continue
    }
//56    continue
    }
//   do 56701 i=1,kraz
    for(i=1;i<=kraz;i++)
    {
        izr=izf[i];
        kcel=kcf[i];
        kdel=kdf[i];

        fprintf(fp9,"%d %d %d\n", izr,kcel,kdel);
        for(ip=1;ip<=kcel;ip++)
            fprintf(fp9,"%d ",icf[(i-1)*k50+ip]);
        fprintf(fp9,"\n");
        for(ip=1;ip<=kdel;ip++)
            fprintf(fp9,"%d ",idf[(i-1)*k50+ip]);
        fprintf(fp9,"\n");
    }
//   close(9,status='keep')
    fclose(fp9);
    kg=k*m+1;
//c   call gay(kg,c,f,ak,igau)

```

```
//c  pause 'pered gaydl'
//   fprintf(fp6, "\npered gaydl\n");
//   if(k==2) return;
//   scanf("%d", &igur);
//   gaydl(kg, izc, kcc, kdc, icc, idc, izf, kcf, kdf, icf, idf,
//   izak, kcak, kdak, icak, idak, &igau, krazr, k50, mnper);
//   *pigau=igau;
//   fprintf(fp6, "\n posle gaydl\n");
//   if(k==2) return;
//   scanf("%d", &igur);
//   if(igau==0) return;
//c  pause 'posle gaydl'
//   kg=k*m+1;
//   fprintf(fp6, "\npered prowg\n");
//   if(k==2) return;
//   prowg(kg, izc, kcc, kdc, icc, idc,
//   izf, kcf, kdf, icf, idf, izak, kcak, kdak, icak, idak, &igau, krazr, k50);
//   fprintf(fp6, "\n posle prowg\n");
//   if(k==2) return;
//   return;
}
```

## Набор gaydl.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

/*
с-метод Гаусса решения систем с выбором главного элемента, длинная арифметика
с-ac(i,j,l)-целая часть элемента i,j матрицы системы
с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
с-nc(I,j)-количество разрядов целой части элемента I,j
с-nd(I,j)-количество разрядов дробной части элемента I,j
с-znak(I,j)-знак элемента I,j
с-bc(I,l)-целая часть правой части i-ого уравнения
с-bd(I,l)-дробная часть правой части i-ого уравнения
с-nbc(i)-длина целой части правой части i-ого уравнения
с-nbd(i)-длина дробной части правой части i-ого уравнения
с-znp(i)-знак правой части i-ого уравнения
с-xc(I,l)-целая часть i-го неизвестного
с-xd(I,l)-дробная часть i-го неизвестного
с-nxc(i)-длина целой части i-ого неизвестного
с-nxd(i)-длина дробной части i-ого неизвестного
с-znx(i)-знак i-ой неизвестной */
void gaydl(int n,int *znak,
int *nc,int *nd,
int *ac,int *ad,
int *znp,int *nbc,int *nbd,
int *bc,int *bd,
int *znx,int *nxc,int *nxd,
int *xc,int *xd,
int *pigau,int krazr,int k50,int *mnper)
{
extern FILE *fp6;

extern void obrdl(int n,int *znak,
int *nc,int *nd,
int *ac,int *ad,
int *znp,int *nbc,int *nbd,
int *bc,int *bd,
int *znx,int *nxc,int *nxd,
int *xc,int *xd,
int *pigau,int *mnper,int anc[400],
int andr[400],int krazr,int k50);

extern void nimax(int i,int j,int n,
int *nc,int *nd, int *znak,
int *ac,int *ad,
int *nbc,int *nbd, int *znp,
int *bc,int *bd,
int *mnper, int krazr,int k50);

extern void delzn(int izr,int iznt,int *pizrez,int ncr,
int ndr,int *rc,int *rd,
int nct,int ndt,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

```

```

extern void ymnzn(int izp,int izr,int *pizrez,int ncp,int ndp,
int *pc,int *pd,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void wiczn(int izr1,int izr,int *pizrez,int ncr1,int ndr1,
int *rc1,int *rd1,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

int anc[400],andr[400],rezc[400],rezd[400],
*pizrez,*pkcrez,*pkdrez,igau,
pc[400],pd[400],rc1[400],rd1[400];
int rc[400],rd[400],tc[400],td[400];
int i,nis,j,nct,ndt,iznt,ld,nolc,nold,iznol,l,ncl,ndl,
iznl,js,izt,nlc,nld,ncr,ndr,izr,kcrez,kdrez,izrez,k,ncp,ndp,
izp,ncr1,ndr1,izr1,zn,ncl,ndl,iznl;

for(i=1;i<=n;i++)
mpper[i]=i;
if(krazr<=k50) goto m8855;
fprintf(fp6," количество разрядов в дробной части>%d,нельзя!\n",k50);
*pigau=0;

return;
m8855:;
nis=n;
i=1;
m81: j=i;
nimax(i,j,n,nc,nd,znak,ac,ad,nbc,nbd,znp,bc,bd,
mpper,krazr,k50);
// fprintf(fp6,"\nposle nimax\n");
//c t=a(i,i)
nct=nc[(i-1l)*n+i];
ndt=nd[(i-1l)*n+i];
iznt=znak[(i-1l)*n+i];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[((i-1l)*n+i-1)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[((i-1l)*n+i-1)*k50+ld];
for(ld=1;ld<=399;ld++)
{
anc[ld]=0;
andr[ld]=0;
}
nolc=399;
nold=399;
iznol=1;
for(ld=1;ld<=nct;ld++)
if(tc[ld]!=anc[ld]) goto m2;
for(ld=1;ld<=ndt;ld++)
if(td[ld]!=andr[ld]) goto m2;
l=i+1;
m4:;
ncl=nc[(l-1l)*n+j];
ndl=nd[(l-1l)*n+j];
iznl=znak[(l-1l)*n+j];
for(ld=1;ld<=ncl;ld++)
if(ac[(l-1l)*n+j-1]*k50+ld]!=anc[ld]) goto m3;
for(ld=1;ld<=ndl;ld++)

```

```

if(ad[(l-1l)*n+j-1]*k50+ld]!=andr[ld]) goto m3;
l=l+1;
if(l<=n) goto m4;
fprintf(fp6," матрица вырожденная \n");

igau=0;
*pigau=igau;
return;
m3:;
for(js=1;js<=n;js++)
{
//c tz=a(l,js)
//c a(l,js)=a(i,js)
//c a(i,js)=tz
nct=nc[(l-1l)*n+js];
ndt=nd[(l-1l)*n+js];
izt=znak[(l-1l)*n+js];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[(l-1l)*n+js-1]*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[(l-1l)*n+js-1]*k50+ld];
nc[(l-1l)*n+js]=nc[(i-1l)*n+js];
nd[(l-1l)*n+js]=nd[(i-1l)*n+js];
znak[(l-1l)*n+js]=znak[(i-1l)*n+js];
n1c=nc[(i-1l)*n+js];
n1d=nd[(i-1l)*n+js];
for(ld=1;ld<=n1c;ld++)
ac[(l-1l)*n+js-1]*k50+ld]=ac[(i-1l)*n+js-1]*k50+ld];
for(ld=1;ld<=n1d;ld++)
ad[(l-1l)*n+js-1]*k50+ld]=ad[(i-1l)*n+js-1]*k50+ld];
nc[(i-1l)*n+js]=nct;
nd[(i-1l)*n+js]=ndt;
znak[(i-1l)*n+js]=izt;
for(ld=1;ld<=nct;ld++)
ac[(i-1l)*n+js-1]*k50+ld]=tc[ld];
for(ld=1;ld<=ndt;ld++)
ad[(i-1l)*n+js-1]*k50+ld]=td[ld];
}
nct=NBC[i];
ndt=NDI[i];
izt=ZNP[i];
for(ld=1;ld<=nct;ld++)
tc[ld]=bc[(i-1l)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=bd[(i-1l)*k50+ld];
NBC[i]=NBC[1];
NDI[i]=NDI[1];
ZNP[i]=ZNP[1];
N1C=NBC[1];
N1D=NDI[1];
for(ld=1;ld<=N1C;ld++)

bc[(i-1l)*k50+ld]=bc[(l-1l)*k50+ld];
for(ld=1;ld<=N1D;ld++)
bd[(i-1l)*k50+ld]=bd[(l-1l)*k50+ld];
NBC[1]=nct;
NDI[1]=ndt;
ZNP[1]=izt;
for(ld=1;ld<=nct;ld++)
bc[(l-1l)*k50+ld]=tc[ld];
for(ld=1;ld<=ndt;ld++)
bd[(l-1l)*k50+ld]=td[ld];

```

```

nct=nc[(i-1l)*n+i];
ndt=nd[(i-1l)*n+i];
iznt=znak[(i-1l)*n+i];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[((i-1l)*n+i-1)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[((i-1l)*n+i-1)*k50+ld];
//c t=a(i,i)
m2;;
m2222;;
//c a(i,j)=a(i,j)/t
ncr=nc[(i-1l)*n+j];
ndr=nd[(i-1l)*n+j];
izr=znak[(i-1l)*n+j];
for(ld=1;ld<=ncr;ld++)
rc[ld]=ac[((i-1l)*n+j-1)*k50+ld];

for(ld=1;ld<=ndr;ld++)
rd[ld]=ad[((i-1l)*n+j-1)*k50+ld];
// fprintf(fp6,"\n прямой ход, деление левой части a(i,j)/t \n");

// fprintf(fp6,"делимое a(i,j)\n");
// fprintf(fp6,"i=%d j=%d\n",i,j);
// fprintf(fp6,"знак=%d\n",izr);
// fprintf(fp6,"длина целой части=%d длина дробной части=%d\n", ncr,ndr);
/* fprintf(fp6,"знак=%d\n",izr);
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n", ncr,ndr);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=ncr;ld++)
fprintf(fp6,"%ld", rc[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=ndr;ld++)
fprintf(fp6,"%ld", rd[ld]);
fprintf(fp6,"\n делитель t \n");
fprintf(fp6,"знак=%d\n",iznt);
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n", nct,ndt);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=nct;ld++)
fprintf(fp6,"%ld", tc[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=ndt;ld++)
fprintf(fp6,"%ld", td[ld]); */
delzn(izr,iznt,&izrez,ncr,ndr,rc,rd,nct,ndt,tc,td,
&kcrez,&kdrez,rezc,rezd,krazr);
/* fprintf(fp6,"\n частное a(i,j)/t \n");
fprintf(fp6,"знак=%d\n",izrez);
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n",
kcrez,kdrez);
fprintf(fp6,"\n целая часть\n");

for(ld=1;ld<=kcrez;ld++)
fprintf(fp6,"%ld", rezc[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=kdrez;ld++)
fprintf(fp6,"%ld", rezd[ld]);

fprintf(fp6,"\nprjm posle delzn a/t\n"); */
nc[(i-1l)*n+j]=kcrez;
nd[(i-1l)*n+j]=kdrez;
znak[(i-1l)*n+j]=izrez;
for(ld=1;ld<=kcrez;ld++)
ac[((i-1l)*n+j-1)*k50+ld]=rezc[ld];

```

```

        for (ld=1;ld<=kdrez;ld++)
        ad[ ((i-1l)*n+j-1)*k50+ld]=rezd[ld];
        j=j+1;
        if (j<=n) goto m2222;
//c   b(i)=b(i)/t
        ncr=ncb[i];
        ndr=ndb[i];
        izr=znb[i];
        for (ld=1;ld<=ncr;ld++)
        rc[ld]=bc[ (i-1l)*k50+ld];
        for (ld=1;ld<=ndr;ld++)
        rd[ld]=bd[ (i-1l)*k50+ld];
        delzn (izr, iznt, &izrez, ncr, ndr, rc, rd, nct, ndt, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
//   fprintf (fp6, "\nprjm posle delzn b/t\n");
        nbc[i]=kcrez;
        nbd[i]=kdrez;
        znp[i]=izrez;
        for (ld=1;ld<=kcrez;ld++)
        bc[ (i-1l)*k50+ld]=rezc[ld];
        for (ld=1;ld<=kdrez;ld++)
        bd[ (i-1l)*k50+ld]=rezd[ld];
        k=i+1;
m71:;
m7111:;
        ncp=nc[ (k-1l)*n+i];
        ndp=nd[ (k-1l)*n+i];
        izp=znak[ (k-1l)*n+i];
        for (ld=1;ld<=ncp;ld++)
        pc[ld]=ac[ ((k-1l)*n+i-1)*k50+ld];
        for (ld=1;ld<=ndp;ld++)
        pd[ld]=ad[ ((k-1l)*n+i-1)*k50+ld];
//c   p=a(k,i)
        j=i;
m7:;
//c a(k,j)=a(k,j)-p*a(i,j)
        ncr=nc[ (i-1l)*n+j];
        ndr=nd[ (i-1l)*n+j];
        izr=znak[ (i-1l)*n+j];
        for (ld=1;ld<=ncr;ld++)
        rc[ld]=ac[ ((i-1l)*n+j-1)*k50+ld];
        for (ld=1;ld<=ndr;ld++)
        rd[ld]=ad[ ((i-1l)*n+j-1)*k50+ld];
        ymnzn (izp, izr, &izrez, ncp, ndp, pc, pd, ncr, ndr, rc, rd,
&kcrez, &kdrez, rezc, rezd, krazr);
//   fprintf (fp6, "\nprjm posle ymnzn p*a(i,j)\n");
        ncr=kcrez;
        ndr=kdrez;
        izr=izrez;
        for (ld=1;ld<=ncr;ld++)
        rc[ld]=rezc[ld];
        for (ld=1;ld<=ndr;ld++)
        rd[ld]=rezd[ld];
        ncr1=nc[ (k-1l)*n+j];
        ndr1=nd[ (k-1l)*n+j];
        izr1=znak[ (k-1l)*n+j];
        for (ld=1;ld<=ncr1;ld++)
        rc1[ld]=ac[ ((k-1l)*n+j-1)*k50+ld];
        for (ld=1;ld<=ndr1;ld++)
        rd1[ld]=ad[ ((k-1l)*n+j-1)*k50+ld];
        wiczn (izr1, izr, &izrez, ncr1, ndr1, rc1, rd1, ncr, ndr, rc,
rd, &kcrez, &kdrez, rezc, rezd, krazr);
//   fprintf (fp6, "\nprjm posle wiczn a(k,j)-p*a(i,j)\n");

```

```

nc[(k-1l)*n+j]=kcrez;
nd[(k-1l)*n+j]=kdrez;
znak[(k-1l)*n+j]=izrez;
for(ld=1;ld<=kcrez;ld++)
ac[(k-1l)*n+j-1]*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
ad[(k-1l)*n+j-1]*k50+ld]=rezd[ld];
j=j+1;
if(j<=n) goto m7;
//c b(k)=b(k)-p*b(i)
ncr=ncb[i];
ndr=ndb[i];
izr=znz[i];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(i-1l)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=bd[(i-1l)*k50+ld];
ymnzn(izp,izr,&izrez,ncp,ndp,pc,pd,ncr,ndr,rc,rd,
&kcrez,&kdrez,rezc,rezd,krazr);
ncr=kcrez;
ndr=kdrez;
izr=izrez;
for(ld=1;ld<=kcrez;ld++)
rc[ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
rd[ld]=rezd[ld];
ncr1=ncb[k];
ndr1=ndb[k];
izr1=znz[k];
for(ld=1;ld<=ncr1;ld++)
rc1[ld]=bc[(k-1l)*k50+ld];
for(ld=1;ld<=ndr1;ld++)
rd1[ld]=bd[(k-1l)*k50+ld];
wiczn(izr1,izr,&izrez,ncr1,ndr1,rc1,rd1,ncr,ndr,rc,rd,
&kcrez,&kdrez,rezc,rezd,krazr);
ncb[k]=kcrez;
ndb[k]=kdrez;
znz[k]=izrez;
for(ld=1;ld<=kcrez;ld++)
bc[(k-1l)*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
bd[(k-1l)*k50+ld]=rezd[ld];
k=k+1;
if(k<=n) goto m71;
i=i+1;
if(i<n) goto m81;
obrdl(n,znak,nc,nd,ac,ad,znz,ncb,ndb,bc,bd,znx,nxc,nxd,
xc,xd,&igau,mnper,anc,andr,krazr,k50);
*pigau=igau;
n=nis;
return;
}

```

**Набор nimax.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnkbc.h"
#include<math.h>
#include<alloc.h>

void nimax(int i,int j,int n,
int *nc ,int *nd , int *znak ,
int *ac ,int *ad ,
int *nbc ,int *nbd , int *znp ,
int *bc ,int *bd ,
int *mnp, int krazr,int k50)
{
int xmaxc[400],xmaxd[400] ,tc[400],td[400],ncel,
ndr,zmax,kcmax,kdmax,ld,imax,jmax,ipoi,jpoi,ncr,
kmin,jper,ktc,ktd,znt,iznr,nlc,nld;
//с-нахождение максимального элемента
//с xmax=dabs(a(i,j))
ncel=nc[(i-1)*n+j];
ndr=nd[(i-1)*n+j];
zmax=1;
kcmax=ncel;
kdmax=ndr;
for(ld=1;ld<=ncel;ld++)
xmaxc[ld]=ac[((i-1)*n+j-1)*k50+ld];
for(ld=1;ld<=ndr;ld++)
xmaxd[ld]=ad[((i-1)*n+j-1)*k50+ld];

imax=i;
jmax=j;
for(ipoi=i;ipoi<=n;ipoi++)
{
for(jpoi=j;jpoi<=n;jpoi++)
{
ncr=nc[(ipoi-1)*n+jpoi];
ndr=nd[(ipoi-1)*n+jpoi];
if(ncr<kcmax) goto m38250;
if(ncr>kcmax) goto m41003;
for(ld=1;ld<=ncr;ld++)
{
if(ac[((ipoi-1)*n+jpoi-1)*k50+ld]>xmaxc[ld]) goto m41003;
if(ac[((ipoi-1)*n+jpoi-1)*k50+ld]<xmaxc[ld]) goto m38250;
}
kmin=kdmax;
if(kmin>ndr) kmin=ndr;
for(ld=1;ld<=kmin;ld++)
{
if(ad[((ipoi-1)*n+jpoi-1)*k50+ld]>xmaxd[ld]) goto m41003;
if(ad[((ipoi-1)*n+jpoi-1)*k50+ld]<xmaxd[ld]) goto m38250;
}
if(kdmax>=ndr) goto m38250;
m41003:;

imax=ipoi;
jmax=jpoi;
kcmax=ncr;
kdmax=ndr;
for(ld=1;ld<=ncr;ld++)
xmaxc[ld]=ac[((ipoi-1)*n+jpoi-1)*k50+ld];

```

```

        for (ld=1; ld<=ndr; ld++)
            xmaxd[ld]=ad[ ((ipoi-1l)*n+jpoi-1)*k50+ld];
m38250:;
    })
//c-пересылка строки i со строкой imax
    for (jper=j; jper<=n; jper++)
    {
//c   t=a(i, jper)
//c   a(i, jper)=a(imax, jper)
//c   a(imax, jper)=t
        ktc=nc[ (i-1l)*n+jper];
        ktd=nd[ (i-1l)*n+jper];
        for (ld=1; ld<=ktc; ld++)
            tc[ld]=ac[ ((i-1l)*n+jper-1)*k50+ld];
        for (ld=1; ld<=ktd; ld++)
            td[ld]=ad[ ((i-1l)*n+jper-1)*k50+ld];
        znt=znak[ (i-1l)*n+jper];
        ncr=nc[ (imax-1l)*n+jper];
        ndr=nd[ (imax-1l)*n+jper];
        for (ld=1; ld<=ncr; ld++)
            ac[ ((i-1l)*n+jper-1)*k50+ld]=ac[ ((imax-1l)*n+jper-1)*k50+ld];
        for (ld=1; ld<=ndr; ld++)
            ad[ ((i-1l)*n+jper-1)*k50+ld]=ad[ ((imax-1l)*n+jper-1)*k50+ld];
        znak[ (i-1l)*n+jper]=znak[ (imax-1l)*n+jper];
        nc[ (i-1l)*n+jper]=ncr;
        nd[ (i-1l)*n+jper]=ndr;
        for (ld=1; ld<=ktc; ld++)
            ac[ ((imax-1l)*n+jper-1)*k50+ld]=tc[ld];
        for (ld=1; ld<=ktd; ld++)
            ad[ ((imax-1l)*n+jper-1)*k50+ld]=td[ld];
        znak[ (imax-1l)*n+jper]=znt;
        nc[ (imax-1l)*n+jper]=ktc;
        nd[ (imax-1l)*n+jper]=ktd;
    }
    ncr=ncb[i];
    ndr=ndb[i];
    iznr=znp[i];
    for (ld=1; ld<=ncr; ld++)
        tc[ld]=bc[ (i-1l)*k50+ld];
    for (ld=1; ld<=ndr; ld++)
        td[ld]=bd[ (i-1l)*k50+ld];
    nbc[i]=ncb[imax];
    nbd[i]=ndb[imax];
    znp[i]=znp[imax];
    n1c=ncb[imax];
    n1d=ndb[imax];
    for (ld=1; ld<=n1c; ld++)
        bc[ (i-1l)*k50+ld]=bc[ (imax-1l)*k50+ld];
    for (ld=1; ld<=n1d; ld++)
        bd[ (i-1l)*k50+ld]=bd[ (imax-1l)*k50+ld];
    nbc[imax]=ncr;
    nbd[imax]=ndr;
    znp[imax]=iznr;
    for (ld=1; ld<=ncr; ld++)
        bc[ (imax-1l)*k50+ld]=tc[ld];
    for (ld=1; ld<=ndr; ld++)
        bd[ (imax-1l)*k50+ld]=td[ld];

//c   t=b(i)
//c   b(i)=b(imax)
//c   b(imax)=t
//c-пересылка столбца j со столбцом jmax
    for (ipoi=1; ipoi<=n; ipoi++)

```

```

    {
//c   t=a(ipoi,j)
//c   a(ipoi,j)=a(ipoi,jmax)
//c   a(ipoi,jmax)=t
        ncr=nc[(ipoi-1l)*n+j];
        ndr=nd[(ipoi-1l)*n+j];
        iznr=znak[(ipoi-1l)*n+j];
        for(ld=1;ld<=ncr;ld++)
            tc[ld]=ac[((ipoi-1l)*n+j-1)*k50+ld];
        for(ld=1;ld<=ndr;ld++)
            td[ld]=ad[((ipoi-1l)*n+j-1)*k50+ld];
        nc[(ipoi-1l)*n+j]=nc[(ipoi-1l)*n+jmax];
        nd[(ipoi-1l)*n+j]=nd[(ipoi-1l)*n+jmax];
        znak[(ipoi-1l)*n+j]=znak[(ipoi-1l)*n+jmax];
        nlc=nc[(ipoi-1l)*n+jmax];
        nld=nd[(ipoi-1l)*n+jmax];
        for(ld=1;ld<=nlc;ld++)
            ac[((ipoi-1l)*n+j-1)*k50+ld]=ac[((ipoi-1l)*n+jmax-1)*k50+ld];
        for(ld=1;ld<=nld;ld++)
            ad[((ipoi-1l)*n+j-1)*k50+ld]=ad[((ipoi-1l)*n+jmax-1)*k50+ld];
        nc[(ipoi-1l)*n+jmax]=ncr;
        nd[(ipoi-1l)*n+jmax]=ndr;
        znak[(ipoi-1l)*n+jmax]=iznr;
        for(ld=1;ld<=ncr;ld++)
            ac[((ipoi-1l)*n+jmax-1)*k50+ld]=tc[ld];
        for(ld=1;ld<=ndr;ld++)
            ad[((ipoi-1l)*n+jmax-1)*k50+ld]=td[ld];
    }
//с-пересылка в массиве номеров mpre столбцов
    jpre=mpre[j];
    mpre[j]=mpre[jmax];
    mpre[jmax]=jpre;
    return;
}

```

## Набор obrdl.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnkbc.h"
#include<math.h>
#include<alloc.h>

/*
с-обратный ход
с-метод Гаусса решения систем с выбором главного элемента, длинная арифметика
с-ac(i,j,l)-целая часть элемента i,j матрицы системы
с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
с-nc(I,j)-количество разрядов целой части элемента I,j
с-nd(I,j)-количество разрядов дробной части элемента I,j
с-знак(I,j)-знак элемента I,j
с-bc(I,l)-целая часть правой части i-ого уравнения
с-bd(I,l)-дробная часть правой части i-ого уравнения
с-nbc(i)-длина целой части правой части i-ого уравнения
с-nbd(i)-длина дробной части правой части i-ого уравнения
с-znp(i)-знак правой части i-ого уравнения
с-xc(I,l)-целая часть i-ого неизвестного
с-xd(I,l)-дробная часть i-ого неизвестного
с-nxc(i)-длина целой части i-ого неизвестного
с-nxd(i)-длина дробной части i-ого неизвестного
с-znx(i)-знак i-ой неизвестной */
void obrdl(int n,int *znak ,
int *nc ,int *nd ,

int *ac ,int *ad ,
int *znp ,int *nbc ,int *nbd ,
int *bc ,int *bd ,
int *znx ,int *nxc ,int *nxd ,
int *xc ,int *xd ,
int *pigau,int *mnper,int anc[400],
int andr[400],int krazr,int k50)
{
extern FILE *fp6;

extern void ymnzn(int izr,int izl,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int ncl,int ndl,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void delzn(int izr,int izt,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int nct,int ndt,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd, int krazr);

extern void slzn(int izsum,int izr,int *pizrez,
int ncsu,m,int ndsum,int *icsu,m,int *idsum,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void wiczn(int izr,int izsum,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int ncsu,m,int ndsum,int *icsu,m,int *idsum,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

int rezc[400],rezd[400],i,*pizrez,*pkcrez,*pkdrez,igau,
icsu,m[400],idsum[400],tc[400],td[400];
int rc[400],rd[400],nis,nct,ndt,izt,ld,nt,ncr,ndr,izr,

```

```

kcrez, kdrez, izrez, ncs, nds, izs, ncl, ndl, izl, j, nlc, nld;
nis=n;
nct=nc[(n-1)*n+n];
ndt=nd[(n-1)*n+n];
izt=znak[(n-1)*n+n];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[(n-1)*n+n-1]*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[(n-1)*n+n-1]*k50+ld];
for(ld=1;ld<=nct;ld++)
if(tc[ld]!=anc[ld]) goto m82;
for(ld=1;ld<=ndt;ld++)
if(td[ld]!=andr[ld]) goto m82;
fprintf(fp6, " матрица вырожденная\n");
igau=0;
*pigau=0;
return;
m82:;
nt=n;
//c x(n)=b(n)/a(n,n)
ncr=nc[n];
ndr=nd[n];
izr=znz[n];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(n-1)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=bd[(n-1)*k50+ld];
delzn(izr, izt, &izrez, ncr, ndr, rc, rd, nct, ndt, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
nxc[n]=kcrez;
nxd[n]=kdrez;
znx[n]=izrez;
for(ld=1;ld<=kcrez;ld++)
xc[(n-1)*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
xd[(n-1)*k50+ld]=rezd[ld];
m85: n=n-1;
if(n<1) goto m83;
//c sum=0
ncs=1;
nds=1;
izs=1;
for(ld=1;ld<=ncs;ld++)
{
ics[ld]=0;
ids[ld]=0;
}
j=n+1;
m84:;
ncr=nc[(n-1)*nt+j];
ndr=nd[(n-1)*nt+j];
izr=znak[(n-1)*nt+j];
for(ld=1;ld<=ncr;ld++)
rc[ld]=ac[(n-1)*nt+j-1]*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=ad[(n-1)*nt+j-1]*k50+ld];
ncl=nxc[j];
ndl=nxd[j];
izl=znz[j];
for(ld=1;ld<=ncl;ld++)
tc[ld]=xc[(j-1)*k50+ld];
for(ld=1;ld<=ndl;ld++)
td[ld]=xd[(j-1)*k50+ld];

```

```

ymnzn(izr, izl, &izrez, ncr, ndr, rc, rd, ncl, ndl, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
ncr=kcrez;
ndr=kdrez;
izr=izrez;
for(ld=1;ld<=ncr;ld++)
rc[ld]=rezc[ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=rezd[ld];
slzn(izsum,izr,&izrez,ncsum,ndsum,icsum,idsum,ncr,ndr,
rc,rd,&kcrez,&kdrez,rezc,rezd,krazr);
ncsum=kcrez;
ndsum=kdrez;
izsum=izrez;
for(ld=1;ld<=kcrez;ld++)
icsum[ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
idsum[ld]=rezd[ld];
//c84 sum=sum+a(n,j)*x(j)
j=j+1;
if(j<=nt) goto m84;
//c x(n)=b(n)-sum
ncr=nb[n];
ndr=nd[n];
izr=znp[n];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(n-1l)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=bd[(n-1l)*k50+ld];
wiczn(izr,izsum,&izrez,ncr,ndr,rc,rd,ncsum,ndsum,icsum,
idsum,&kcrez,&kdrez,rezc,rezd,krazr);
nxc[n]=kcrez;
nxd[n]=kdrez;
znx[n]=izrez;
for(ld=1;ld<=kcrez;ld++)
xc[(n-1l)*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
xd[(n-1l)*k50+ld]=rezd[ld];
goto m85;
m83:;
igau=1;
*pigau=igau;
//с-расстановка значений в векторе решения X
for(i=1;i<=nt;i++)
{
j=mp[er[i];
//c b(j)=x(i)
nb[j]=nxc[i];
nd[j]=nxd[i];
znp[j]=znx[i];
n1c=nxc[i];
n1d=nxd[i];
for(ld=1;ld<=n1c;ld++)
bc[(j-1l)*k50+ld]=xc[(i-1l)*k50+ld];
for(ld=1;ld<=n1d;ld++)
bd[(j-1l)*k50+ld]=xd[(i-1l)*k50+ld];
}
for(i=1;i<=nt;i++)
{
//c x(i)=b(i)
nxc[i]=nb[i];
nxd[i]=nd[i];
znx[i]=znp[i];

```

```
n1c=nbk[i];
n1d=nbd[i];
for (ld=1;ld<=n1c;ld++)
xc[(i-11)*k50+ld]=bc[(i-11)*k50+ld];
for (ld=1;ld<=n1d;ld++)
xd[(i-11)*k50+ld]=bd[(i-11)*k50+ld];
}
n=nis;
return;
}
```

## Набор prowg.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

        void prowg(int n,int *znak ,
        int *nc ,int *nd ,
        int *ac ,int *ad ,
        int *znp ,int *nbc ,int *nbd ,
        int *bc ,int *bd ,
        int *znx ,int *nxc ,int *nxd ,
        int *xc ,int *xd ,
        int *pigau,int krazr,int k50)
{
    extern void ymnzn(int izn1,int izn2,
int *pizrez,int ikc1,int ikd1,int *ic1,int *id1,
int ikc2,int ikd2,int *ic2,int *id2,
    int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void slzn(int izsum,int izn2,int *pizrez,int isc,int isd,
int *icsum,int *idsum,
int ikc2,int ikd2,
    int *ic2,int *id2,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void wiczn(int izsum,int izb,int *pizrez,int isc,int isd,
int *icsum,int *idsum,int icb,int idb,int *ic1,int *id1,
    int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern FILE *fp5,*fp9,*fp6;

//с-проверка метода Гаусса
//с-ac(i,j,l)-целая часть элемента i,j матрицы системы
//с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
//с-nc(I,j)-количество разрядов целой части элемента I,j
//с-nd(I,j)-количество разрядов дробной части элемента I,j
//с-znak(I,j)-знак элемента I,j
//с-bc(I,l)-целая часть правой части i-ого уравнения
//с-bd(I,l)-дробная часть правой части i-ого уравнения
//с-nbc(i)-длина целой части правой части i-го уравнения
//с-nbd(i)-длина дробной части правой части i-го уравнения
//с-znp(i)-знак правой части i-ого уравнения
//с-xc(I,l)-целая часть i-го неизвестного
//с-xd(I,l)-дробная часть i-го неизвестного
//с-nxc(i)-длина целой части i-ого неизвестного
//с-nxd(i)-длина дробной части i-ого неизвестного
//с-znx(i)-знак i-ой неизвестной
    int icel[400],idd[400],*pizrez,*pkcrez,*pkdrez,
    ic1[400],id1[400],
    ic2[400],id2[400],icrez[400],idrez[400],zn,l,
    icsum[400],idsum[400],rabl;
    int i,j,ncz,ndz,nbcz,nbdz,isc,isd,izn2,ikc2,ikd2,ld,izn1,ikc1,
    ikd1,izrez,kcrez,kdrez,izsum,izb,icb,idb;
    fp9=fopen("promprc","r");
    fscanf(fp9,"%d",&n);
    fprintf(fp5,"\nпроверка решения,матрица системы\n");
    for(i=1;i<=n;i++)
    {

```

```

for(j=1;j<=n;j++)
{
fprintf(fp5, "\nэлемент с(%d,%d)\n", i, j);
fscanf(fp9, "%d%d%d", &rabl, &ncz, &ndz);
znak[(i-1l)*n+j]=rabl;
for(l=1;l<=ncz;l++)
{
fscanf(fp9, "%d", &rabl);
ac[((i-1l)*n+j-1)*k50+1]=rabl;
}
for(l=1;l<=ndz;l++)
{
fscanf(fp9, "%d", &rabl);
ad[((i-1l)*n+j-1)*k50+1]=rabl;
}
fprintf(fp5, "\nзнак числа=%2d\n", znak[(i-1l)*n+j]);
fprintf(fp5, "\nколичество разрядов в целой=%3d, в дробной=%3d\n",
ncz, ndz);
fprintf(fp5, "\nцелая часть числа\n");
for(ld=1;ld<=ncz;ld++)
fprintf(fp5, "%1d", ac[((i-1l)*n+j-1)*k50+ld]);
fprintf(fp5, "\ndробная часть числа\n");
for(ld=1;ld<=ndz;ld++)
fprintf(fp5, "%1d", ad[((i-1l)*n+j-1)*k50+ld]);
fprintf(fp5, "\n");
nc[(i-1l)*n+j]=ncz;
nd[(i-1l)*n+j]=ndz;
}
}
// fprintf(fp6, "\n проwg после wiw matr\n");
// if(n==3) return;
fprintf(fp5, "\n свободные члены\n");
for(i=1;i<=n;i++)
{
fprintf(fp5, "\n i=%3d\n", i);
fscanf(fp9, "%d%d%d", &znp[i], &nbcz, &nbdz);
for(l=1;l<=nbcz;l++)
{
fscanf(fp9, "%d", &rabl);
bc[(i-1l)*k50+1]=rabl;
}
for(l=1;l<=nbdz;l++)
{
fscanf(fp9, "%d", &rabl);
bd[(i-1l)*k50+1]=rabl;
}
fprintf(fp5, "\nзнак свободного члена %3d\n", znp[i]);
fprintf(fp5, "\nколичество разрядов в целой части=%3d, в дробной=%3d\n",
nbcz, nbdz);
fprintf(fp5, "\nцелая часть\n");
for(ld=1;ld<=nbcz;ld++)
fprintf(fp5, "%1d", bc[(i-1l)*k50+ld]);
fprintf(fp5, "\ndробная часть\n");
for(ld=1;ld<=nbdz;ld++)
fprintf(fp5, "%1d", bd[(i-1l)*k50+ld]);
fprintf(fp5, "\n");
nbc[i]=nbcz;
nbd[i]=nbdz;
}
fprintf(fp5, "\n неизвестные системы\n");
for(i=1;i<=n;i++)
{
fprintf(fp5, "\nномер неизвестного-%3d\n", i);

```

```

nbcz=nxc[i];
nbdz=nxd[i];
fprintf(fp5, "\nзнак неизвестного %2d\n", znx[i]);
fprintf(fp5, "количество разрядов в целой части=%3d, в дробной=%3d\n",
nbcz, nbdz);
fprintf(fp5, "\nцелая часть\n");
for(ld=1; ld<=nbcz; ld++)
fprintf(fp5, "%1d", xc[(i-1l)*k50+ld]);
fprintf(fp5, "\ndробная часть\n");
for(ld=1; ld<=nbdz; ld++)
fprintf(fp5, "%1d", xd[(i-1l)*k50+ld]);
fprintf(fp5, "\n");
}
// fprintf(fp5, "\n peredras prowg\n");

// if(n==3) return;
for(i=1; i<=n; i++)
{
//c sum=0
izsum=1;
isc=1;
isd=1;
icsum[1]=0;
idsum[1]=0;
for(j=1; j<=n; j++)
{
izn2=znx[j];
ikc2=nxc[j];
ikd2=nxd[j];
for(ld=1; ld<=ikc2; ld++)
ic2[ld]=xc[(j-1l)*k50+ld];
for(ld=1; ld<=ikd2; ld++)
id2[ld]=xd[(j-1l)*k50+ld];
izn1=znak[(i-1l)*n+j];
ikc1=nc[(i-1l)*n+j];

ikd1=nd[(i-1l)*n+j];
for(ld=1; ld<=ikc1; ld++)
ic1[ld]=ac[((i-1l)*n+j-1)*k50+ld];
for(ld=1; ld<=ikd1; ld++)
id1[ld]=ad[((i-1l)*n+j-1)*k50+ld];

ymnzn(izn1, izn2, &izrez, ikc1, ikd1, ic1, id1, ikc2, ikd2, ic2, id2,
&kcrez, &kdrez, icrez, idrez, krazr);
izn2=izrez;
ikc2=kcrez;
ikd2=kdrez;
for(ld=1; ld<=kcrez; ld++)
m323: ic2[ld]=icrez[ld];
for(ld=1; ld<=kdrez; ld++)
m423: id2[ld]=idrez[ld];
slzn(izsum, izn2, &izrez, isc, isd, icsum, idsum, ikc2, ikd2,
ic2, id2, &kcrez, &kdrez, icrez, idrez, krazr);

izsum=izrez;
isc=kcrez;
isd=kdrez;
for(ld=1; ld<=kcrez; ld++)
m555: icsum[ld]=icrez[ld];
for(ld=1; ld<=kdrez; ld++)
m666: idsum[ld]=idrez[ld];
}
izb=znz[i];

```

```

        icb=NBC[i];
        idb=nbD[i];
        for (ld=1;ld<=icb;ld++)
m815: ic1[ld]=bc[(i-1)*k50+ld];
        for (ld=1;ld<=idb;ld++)
m816: id1[ld]=bd[(i-1)*k50+ld];
        wiczn(izsum,izb,&izrez,isc,isd,icsum,idsum,icb,idb,ic1,id1,
&kcrez,&kdrez,icrez,idrez,krazr);
        fprintf(fp5,"\n");
        fprintf(fp5,"\n уравнение-%3d\n",i);
        fprintf(fp5,"\n");
        fprintf(fp5,"\n правая часть,исход\n");
        fprintf(fp5,"\n знак=%2d\n",izb);
        fprintf(fp5,"\n длина целой=%3d длина дробной=%3d\n", icb,idb );
        fprintf(fp5,"\n целая часть=\n");
        for (ld=1;ld<=icb;ld++)
        fprintf(fp5,"%1d", ic1[ld]);
        fprintf(fp5,"\n дробная часть=\n");
        for (ld=1;ld<=idb;ld++)
        fprintf(fp5,"%1d", id1[ld]);
        fprintf(fp5,"\n");

        fprintf(fp5,"\n правая часть,счет\n");
        fprintf(fp5,"\n знак=%2d\n",izsum);
        fprintf(fp5,"\n длина целой=%3d длина дробной=%3d\n", isc,isd);
        fprintf(fp5,"\n целая часть\n");
        for (ld=1;ld<=isc;ld++)
        fprintf(fp5,"%1d", icsum[ld]);
        fprintf(fp5,"\n дробная часть\n");
        for (ld=1;ld<=isd;ld++)
        fprintf(fp5,"%1d", idsum[ld]);
        fprintf(fp5,"\n");
        fprintf(fp5,"\n разница\n");
        fprintf(fp5,"\n знак=%2d\n",izrez);
        fprintf(fp5,"\n длина целой=%3d длина дробной=%3d\n", kcrez,kdrez);
        fprintf(fp5,"\n целая часть\n");
        for (ld=1;ld<=kcrez;ld++)
        fprintf(fp5,"%1d", icrez[ld]);
        fprintf(fp5,"\n дробная часть\n");
        for (ld=1;ld<=kdrez;ld++)
        fprintf(fp5,"%1d", idrez[ld]);
        fprintf(fp5,"\n");
    }
    fclose(fp9);
// fclose(fp5);
    return;
}

```

**Набор sl.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnkbc.h"
#include<math.h>
#include<alloc.h>

//с-длинное сложение
void sl(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{
int c[401],cd[401],i,j,ir,max,in,kcel,kdro,
it,ik,p,nt,mt,rez,il,os;

for(i=1;i<=399;i++)
{
rezc[i]=0;
rezdr[i]=0;
}
if(n>=m) goto m897;
for(i=1;i<=n;i++)
c[i]=a[i];
for(i=1;i<=nd;i++)
cd[i]=ad[i];
for(i=1;i<=m;i++)
a[i]=b[i];
for(i=1;i<=md;i++)
ad[i]=bd[i];
for(i=1;i<=n;i++)
b[i]=c[i];
for(i=1;i<=nd;i++)
bd[i]=cd[i];
ir=n;
n=m;
m=ir;
ir=nd;
nd=md;
md=ir;
m897:;
//с-формирование двух массивов для сложения
max=nd;
if(max<md) max=md;
if(md==max) goto m78;
in=md+1;
for(i=in;i<=max;i++)
bd[i]=0;
goto m81;
m78:;
in=nd+1;
for(i=in;i<=max;i++)
ad[i]=0;
m81:;
it=0;
in=n+1;
ik=n+max;
for(i=in;i<=ik;i++)
{
it=it+1;
a[i]=ad[it];
}
}

```

```

    it=0;
    in=m+1;
    ik=m+max;
    for (i=in; i<=ik; i++)
    {
        it=it+1;
        b[i]=bd[it];
    }

//с-сложение
    i=0;
    p=0;
    nt=n+max;
    mt=m+max;
m7:;
    if (mt<1) rez=a[nt];
    if (mt>=1) rez=a[nt]+b[mt];
    rez=rez+p;
    i1=rez/10;
    os=rez-i1*10;
    i=i+1;
    c[i]=os;
//c
    if (i.ne.1) goto 1
    p=i1;
    nt=nt-1;
    mt=mt-1;
    if ((p==0) && (mt<1)) goto m4;
    if ((nt<1) && (p!=0)) goto m5;
    if ((nt<1) && (p==0)) goto m4;
    goto m7;
m5:;
    i=i+1;
    c[i]=p;
    goto m67;
m4:;

m89:;
    if (nt<1) goto m67;
    i=i+1;
    c[i]=a[nt];
    nt=nt-1;
    goto m89;
m67:;
    ik=max+1;
    kcel=i-max;
    kdro=max;
    it=0;
    for (j=i; j>=ik; j=j-1)
    {
        it=it+1;
        rezc[it]=c[j];
    }
    it=0;
    for (j=max; j>=1; j=j-1)
    {
        it=it+1;
        rezdr[it]=c[j];
    }
    if (kcel!=0) goto m971;
    kcel=1;
    rezc[1]=0;
m971:;
    if (kdro!=0) goto m972;

```

```
    kdro=1;
    rezdr[1]=0;
m972:;
    if(kdro>krazr) kdro=krazr;
    *pkcel=kcel;
    *pkdro=kdro;
    return;
}
```

**Набор slzn.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

void slzn(int za,int zb,int *pzc,
int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{

extern void sl(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

extern void wich(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

//с-длинное сложение с учетом знаков слагаемых

int c[401],cd[401],kcel,kdro,
ar[401],adr[401],zc,
ai[401],adi[401],bi[401],bdi[401];
int zais,zbis,i,ni,ndi,mi,mdi,ip,nmin,nr,ndr;
zais=za;
zbis=zb;
for(i=1;i<=n;i++)
ai[i]=a[i];
for(i=1;i<=nd;i++)
adi[i]=ad[i];
for(i=1;i<=m;i++)
bi[i]=b[i];
for(i=1;i<=md;i++)
bdi[i]=bd[i];
ni=n;
ndi=nd;
mi=m;
mdi=md;

ip=za*zb;
if(ip>0) goto m781;
if(za==1) goto m782;
m1901: if(m>n) goto m783;
if(m==n) goto m784;
wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
*pzc=za;
goto m785;
m784:;
for(i=1;i<=m;i++)
{
if(a[i]>b[i]) goto m787;
if(a[i]<b[i]) goto m788;
}
nmin=nd;
if(nmin>md) nmin=md;
for(i=1;i<=nmin;i++)
{
if(ad[i]>bd[i]) goto m792;
}
}

```

```

    if(ad[i]<bd[i]) goto m793;
  }
  if(nd<=md) goto m795;
  wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
  goto m785;
m795:;
  wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=zb;
  goto m785;
m793:;
  wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=zb;
  goto m785;
m792:;
  wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
  goto m785;
m788:;
  wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=zb;
  goto m785;
m787:;
  wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
  goto m785;
m783:;
  wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=zb;
  goto m785;
m782:;
  for(i=1;i<=n;i++)
    ar[i]=a[i];

  for(i=1;i<=nd;i++)
    adr[i]=ad[i];
  nr=n;
  ndr=nd;
  for(i=1;i<=m;i++)
    a[i]=b[i];
  for(i=1;i<=md;i++)
    ad[i]=bd[i];
  n=m;
  nd=md;
  m=nr;
  md=ndr;
  for(i=1;i<=nr;i++)
    b[i]=ar[i];
  for(i=1;i<=ndr;i++)
    bd[i]=adr[i];
  zc=za;
  za=zb;
  zb=zc;
  goto m1901;
m781:;
  sl(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
m785:;
  n=ni;
  nd=ndi;
  m=mi;
  md=mdi;
  for(i=1;i<=n;i++)

```

```
a[i]=ai[i];
for(i=1;i<=nd;i++)
ad[i]=adi[i];
for(i=1;i<=m;i++)
b[i]=bi[i];
for(i=1;i<=md;i++)
bd[i]=bdi[i];

za=zais;
zb=zbis;
*pkcel=kcel;
*pkdro=kdro;
return;
}
```

**Набор wich.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnkbc.h"
#include<math.h>
#include<alloc.h>

//с-длинное вычитание
void wich(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcra,int *pkdra,int *cra,int *dra,int krazr)
{
int c[401],cd[401],i,ir,max,jp,kcra,kdra,
in,it,ik,nt,mt,np,il,j,im;

for(i=1;i<=399;i++)
{
cra[i]=0;
dra[i]=0;
}
if(n>=m) goto m897;
for(i=1;i<=n;i++)
c[i]=a[i];
for(i=1;i<=nd;i++)
cd[i]=ad[i];
for(i=1;i<=m;i++)
a[i]=b[i];
for(i=1;i<=md;i++)
ad[i]=bd[i];
for(i=1;i<=n;i++)
b[i]=c[i];
for(i=1;i<=nd;i++)
bd[i]=cd[i];
ir=n;
n=m;
m=ir;
ir=nd;
nd=md;
md=ir;
m897:;

//с-формирование двух массивов для вычитания
max=nd;
if(max<md) max=md;
if(md==max) goto m78;
in=md+1;
for(i=in;i<=max;i++)
bd[i]=0;
goto m81;
m78:;
in=nd+1;
for(i=in;i<=max;i++)
ad[i]=0;
m81:;
it=0;
in=n+1;
ik=n+max;
for(i=in;i<=ik;i++)
{
it=it+1;
a[i]=ad[it];
}

```

```

    }
    it=0;
    in=m+1;
    ik=m+max;
    for (i=in; i<=ik; i++)
    {
        it=it+1;
        b[i]=bd[it];
    }
    i=0;
    nt=n+max;
    mt=m+max;
m10:;
    if (a[nt]>=b[mt]) goto m1;
    np=nt-1;
m3:;
    if (np==0) goto m201;
    if (a[np]!=0) goto m2;
    np=np-1;
    goto m3;
m2:;
    a[np]=a[np]-1;
m7:;
    np=np+1;
    if (np==nt) goto m30;
    a[np]=9;
    goto m7;
m30:;
    a[nt]=a[nt]+10;
    goto m10;
m1:;
    i=i+1;
    c[i]=a[nt]-b[mt];
    nt=nt-1;
    mt=mt-1;
    if (mt!=0) goto m10;
    if (n==m) goto m781;
m561:;
    i=i+1;
    c[i]=a[nt];
    nt=nt-1;
    if (nt>=1) goto m561;
m781:;
    in=i;
    ik=max+1;
    in=max;
    goto m999;
m201:;
// write(5,*) 'из меньшего вычитаем большее,здесь нельзя!'
// exit(0);
return;
m999:;
    kcra=i-max;
    kdra=max;
    it=0;
    in=i;
    ik=max+1;
    for (j=i; j>=ik; j=j-1)
    {
        it=it+1;
        cra[it]=c[j];
    }
    im=0;

```

```

    for(j=1;j<=it;j++)
    {
    if(cra[j]!=0) goto m3122;
    im=im+1;
    }
    goto m973;
m3122:;
    il=0;
    for(jp=j;jp<=it;jp++)
    {
    il=il+1;
    cra[il]=cra[jp];
    }
m973:;
    kcra=kcra-im;
    it=0;
    in=max;
    for(j=max;j>=1;j=j-1)
    {
    it=it+1;
    dra[it]=c[j];
    }
    if(kcra!=0) goto m971;
    kcra=1;
    cra[1]=0;
m971:;
    if(kdra!=0) goto m972;
    kdra=1;
    dra[1]=0;
m972:;
    if(kdra>krazr) kdra=krazr;
    *pkcra=kcra;
    *pkdra=kdra;
    return;
    }

```

**Набор wiczn.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

void wiczn(int za,int zb,int *pzc,
int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{
extern void sl(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

extern void wich(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

// с-длинное вычитание с учетом знаков уменьшаемого и вычитаемого
int c[401],cd[401],kcel,kdro,zc,
ar[401],adr[401],
ai[401],adi[401],bi[401],bdi[401],zais,zbi,ip,
zbis,i,ni,ndi,mi,mdi,nmin,nr,ndr;

zais=za;
zbis=zb;
for(i=1;i<=n;i++)
ai[i]=a[i];
for(i=1;i<=nd;i++)
adi[i]=ad[i];
for(i=1;i<=m;i++)
bi[i]=b[i];
for(i=1;i<=md;i++)
bdi[i]=bd[i];
ni=n;
ndi=nd;
mi=m;
mdi=md;
zbi=zb;
zb=zb*(-1);
ip=za*zbi;
if(ip>0) goto m781;
if(za==1) goto m782;
m1901:if(m>n) goto m783;
if(m==n) goto m784;
wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
*pzc=za;
goto m785;
m784:;
for(i=1;i<=m;i++)
{
if(a[i]>b[i]) goto m787;
if(a[i]<b[i]) goto m788;
}
nmin=nd;
if(nmin>md) nmin=md;
for(i=1;i<=nmin;i++)
{
if(ad[i]>bd[i]) goto m792;
}
}

```

```

    if(ad[i]<bd[i]) goto m793;
    }
    if(nd<=md) goto m795;
    wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=za;
    goto m785;
m795:;
    wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=zb;
    goto m785;
m793:;
    wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=zb;
    goto m785;
m792:;
    wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=za;
    goto m785;
m788:;
    wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=zb;
    goto m785;
m787:;
    wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=za;
    goto m785;
m783:;
    wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=zb;
    goto m785;
m782:;
    for(i=1;i<=n;i++)
    ar[i]=a[i];
    for(i=1;i<=nd;i++)
    adr[i]=ad[i];
    nr=n;
    ndr=nd;
    for(i=1;i<=m;i++)
    a[i]=b[i];
    for(i=1;i<=md;i++)
    ad[i]=bd[i];
    n=m;
    nd=md;
    m=nr;
    md=ndr;
    for(i=1;i<=nr;i++)
    b[i]=ar[i];
    for(i=1;i<=ndr;i++)
    bd[i]=adr[i];
    zc=za;
    za=zb;
    zb=zc;
    goto m1901;
m781:;
    sl(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=za;
m785:;
    zb=zbi;
    n=ni;
    nd=ndi;
    m=mi;
    md=mdi;
    for(i=1;i<=n;i++)

```

```
a[i]=ai[i];
for(i=1;i<=nd;i++)
ad[i]=adi[i];
for(i=1;i<=m;i++)
b[i]=bi[i];
for(i=1;i<=md;i++)
bd[i]=bdi[i];
za=zais;
zb=zbis;
*pkcel=kcel;
*pkdro=kdro;
return;
}
```

**Набор ymnzn.cpp**

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

void ymnzn(int za,int zb,int *pzc,int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr)
{

extern void ymn(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr);

//с-длинное умножение с учетом знаков сомножителей
int c[400],cd[400],kcel,kdro,zc,
csl[400],csld[400],
rezc[400],rezdr[400],
ai[400],adi[400],bi[400],bdi[400];
int zais,zbis,i,ni,ndi,mi,mdi;
zais=za;
zbis=zb;
for(i=1;i<=n;i++)
ai[i]=a[i];
for(i=1;i<=nd;i++)
adi[i]=ad[i];
for(i=1;i<=m;i++)
bi[i]=b[i];
for(i=1;i<=md;i++)
bdi[i]=bd[i];
ni=n;
ndi=nd;
mi=m;
mdi=md;
*pzc=za*zb;

ymn(n,nd,a,ad,m,md,b,bd,pkcel,pkdro,ctot,cdtot,krazr);

n=ni;
nd=ndi;
m=mi;
md=mdi;
for(i=1;i<=n;i++)
a[i]=ai[i];
for(i=1;i<=nd;i++)
ad[i]=adi[i];
for(i=1;i<=m;i++)
b[i]=bi[i];
for(i=1;i<=md;i++)
bd[i]=bdi[i];
za=zais;
zb=zbis;

return;
}

```

**//с-длинное умножение умн**

```

void ymn(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr)
{

extern void sl(int icel,int idr,int *csl,int *csld,
int ntot,int kdtot,int *ctot,int *cdtot,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

int c[400],cd[400],i,it,in,ik,ntot,kdtot,im,p,nt,mt,jp,pr,il,os,
kraz,kcd,icel,iab,idr,j,il,kcel,kdro,
csl[400],csld[400],
rezc[400],rezdr[400];
for(i=1;i<=399;i++)
{
ctot[i]=0;
cdtot[i]=0;
}
//с-формирование двух массивов для умножения
it=0;
in=n+1;
ik=n+nd;
for(i=in;i<=ik;i++)
{
it=it+1;
a[i]=ad[it];
}
it=0;
in=m+1;
ik=m+md;
for(i=in;i<=ik;i++)
{
it=it+1;
b[i]=bd[it];
}
ntot=n+m;
kdtot=nd+md;
for(i=1;i<=ntot;i++)
ctot[i]=0;
for(i=1;i<=kdtot;i++)
cdtot[i]=0;

//с-умножение
i=0;
im=0;
p=0;
nt=n+nd;
mt=m+md;
m801:;
p=0;
for(jp=1;jp<=399;jp++)
c[jp]=0;
i=0;
nt=n+nd;
im=im+1;
m701:;
pr=a[nt]*b[mt]+p;
il=pr/10;

```

```

os=pr-i1*10;
i=i+1;
c[i]=os;
p=i1;
nt=nt-1;
if(nt>=1) goto m701;
if(p==0) goto m1701;
i=i+1;
c[i]=p;
p=0;
m1701:;
kraz=md-im+1;
kcd=nd+kraz;
icel=i-kcd;
if(icel<i) goto m1921;
iab=kcd;
if(kcd<0) iab=-kcd;
it=0;
for(jp=i;jp>=1;jp=jp-1)
{
it=it+1;
csl[it]=c[jp];
}
if(iab==0) goto m1941;
for(jp=iab;jp>=1;jp=jp-1)
{
it=it+1;
csl[it]=0;
}
m1941:;
idr=1;
csld[1]=0;
goto m8722;
m1921:;
it=0;
in=i;
ik=kcd+1;
if(in==ik) goto m961;
if(in<ik) goto m962;
for(j=in;j>=ik;j=j-1)
{
it=it+1;
csl[it]=c[j];
}
icel=i-kcd;
goto m963;
m961:;
icel=1;
csl[1]=c[in];
goto m963;
m962:;
icel=1;
csl[1]=0;
m963:;
in=kcd;
it=0;
for(j=in;j>=1;j=j-1)
{
it=it+1;
csld[it]=c[j];
}
idr=kcd;
if(idr>0) goto m8722;

```

```

        idr=1;
        csld[1]=0;
m8722:;

        sl(icel,idr,cs1,csld,ntot,kdtot,ctot,cdtot,&kcel,&kdro,
        rezc,rezdr,krazr);
        if(kcel!=0) goto m8723;
        kcel=1;
        rezc[1]=0;
m8723:;
        if(kdro!=0) goto m8724;
        kdro=1;
        rezdr[1]=0;
m8724:;
        for(i=1;i<=kcel;i++)
            ctot[i]=rezc[i];
        for(i=1;i<=kdro;i++)
            cdtot[i]=rezdr[i];
        ntot=kcel;
        kdtot=kdro;
        mt=mt-1;
        if(mt>=1) goto m801;
        it=0;
        for(j=1;j<=kcel;j++)
        {
            if(ctot[j]!=0) goto m7767;
            it=it+1;
        }
m7767:;
        il=0;
        for(jp=j;jp<=kcel;jp++)
        {
            il=il+1;
            ctot[il]=ctot[jp];
        }
        kcel=kcel-it;
        if(kcel!=0) goto m3999;
        kcel=1;
        ctot[1]=0;
m3999:;
        if(kdro!=0) goto m4000;
        kdro=1;
        cdtot[1]=0;
m4000:;

        if(kdro>krazr) kdro=krazr;
        *pkcel=kcel;
        *pkdro=kdro;
        return;

    }

```

## Набор delzn.cpp

```

#include<windows.h>
#include<string.h>
#include<stdio.h>
#include"dlnmkbc.h"
#include<math.h>
#include<alloc.h>

//с-длинное деление с учетом знаков
void delzn(int za,int zb,int *pzc,
    int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr)
{
    extern FILE *fp6;
    extern void deld(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr);

    int c[400],cd[400],ni,ndi,mi,mdi,kcdel,kddel,
    ar[400],adr[400],br[400],bdr[400],zais,zbis,i,
    ctot[400],cdtot[400],ld,
    rezc[400],rezdr[400],ce[400],ced[400];
    zais=za;
    zbis=zb;
    for(i=1;i<=n;i++)
    ar[i]=a[i];
    for(i=1;i<=nd;i++)
    adr[i]=ad[i];
    for(i=1;i<=m;i++)
    br[i]=b[i];
    for(i=1;i<=md;i++)
    bdr[i]=bd[i];
    ni=n;
    ndi=nd;
    mi=m;
    mdi=md;
    *pzc=za*zb;
/*    fprintf(fp6,"делимое delzn\n");
    fprintf(fp6,"длина целой части=%d длина дробной части=%d\n", n,nd);
    fprintf(fp6,"\n целая часть\n");
    for(ld=1;ld<=n;ld++)
    fprintf(fp6,"%1d", a[ld]);
    fprintf(fp6,"\n дробная часть\n");
    for(ld=1;ld<=nd;ld++)
    fprintf(fp6,"%1d", ad[ld]);
    fprintf(fp6,"\n делитель \n");
//    fprintf(fp6,"знак=%d\n",iznt);
    fprintf(fp6,"длина целой части=%d длина дробной части=%d\n",m,md);
    fprintf(fp6,"\n целая часть\n");
    for(ld=1;ld<=m;ld++)
    fprintf(fp6,"%1d", b[ld]);
    fprintf(fp6,"\n дробная часть\n");
    for(ld=1;ld<=md;ld++)
    fprintf(fp6,"%1d", bd[ld]);
    fprintf(fp6,"\n"); */

    deld(n,nd,a,ad,m,md,b,bd,&kcdel,&kddel,cdel,ddel,krazr);
/*    fprintf(fp6,"частное delzn\n");

```

```

fprintf(fp6,"длина целой части=%d длина дробной части=%d\n",kcdel,kddel);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=kcdel;ld++)
fprintf(fp6,"%ld", cdel[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=kddel;ld++)
fprintf(fp6,"%ld", ddel[ld]);
fprintf(fp6,"\n");          */
n=ni;
nd=ndi;
m=mi;
md=mdi;
for(i=1;i<=n;i++)
a[i]=ar[i];
for(i=1;i<=nd;i++)
ad[i]=adr[i];
for(i=1;i<=m;i++)
b[i]=br[i];
for(i=1;i<=md;i++)
bd[i]=bdr[i];
za=zais;
zb=zbis;
*pkcdel=kcdel;
*pkddel=kddel;
return;
}
//deld
void deld(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr)
{
extern FILE *fp6;
extern void wich(int nat,int kad,int *a,int *ad,
int mat,int kbd,int *b,int *bd,
int *pkc,int *pkcd,int *c,int *cd,int krazr);

extern void sl(int kctot,int kdto,int *ctot,int *cdtot,
int kedi,int kedid,int *ce,int *ced,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

//с-длинное деление
int c[400],cd[400],kcdel,kddel,
ctot[400],cdtot[400],*pkc,*pkcd,*pkcel,*pkdro,
rezc[400],rezdr[400],ce[400],ced[400];
int i,max,in,it,ik,il,j,ill,jp,kctot,kdto,
kedi,kedid,kad,kbd,nat,mat,isd,kcel,kdro,kc,kcd;

for(i=1;i<=399;i++)
{
cdel[i]=0;
ddel[i]=0;
}
//с-формирование двух массивов для деления
max=nd;
if(max<md) max=md;
if(md==max) goto m78;
in=md+1;
for(i=in;i<=max;i++)
bd[i]=0;
goto m81;
m78:;
in=nd+1;

```

```

        for (i=in; i<=max; i++)
            ad[i]=0;
m81:;
    it=0;
    in=n+1;
    ik=n+max;
    for (i=in; i<=ik; i++)
    {

        it=it+1;
        a[i]=ad[it];
    }

    il=0;
    for (j=1; j<=ik; j++)
    {

        if (a[j]!=0) goto m5672;
        il=il+1;
    }
    kcdel=0;
    kddel=0;

    goto m1231;
//    exit(0);
m5672:;
    it=0;
    for (jp=j; jp<=ik; jp++)
    {

        it=it+1;
        a[it]=a[jp];
    }

    it=0;
    in=m+1;
    ik=m+max;
    for (i=in; i<=ik; i++)
    {

        it=it+1;
        b[i]=bd[it];
    }

    ill=0;
    for (j=1; j<=ik; j++)
    {

        if (b[j]!=0) goto m15672;
        ill=ill+1;
    }
    fprintf(fp6, "на ноль делить нельзя!\n");
    kcdel=0;
    kddel=0;
    goto m1231;
//    exit(0);
m15672:;
    it=0;
    for (jp=j; jp<=ik; jp++)
    {

        it=it+1;
        b[it]=b[jp];

```

```

    }

//c      if(n.lt.m) goto 675
//c      if(n.eq.m) goto 676
      for(i=1;i<=399;i++)
      {
        ctot[i]=0;
        cdtot[i]=0;
      }
      kctot=1;
      kdto=1;
      ce[1]=1;
      kedi=1;
      ced[1]=0;
      kedid=1;
      kad=1;
      kbd=1;
      ad[1]=0;
      bd[1]=0;
      nat=n+max-11;
      mat=m+max-111;
      isd=0;
      kcel=1;
      kdro=1;
      rezc[1]=0;
      rezdr[1]=0;
m2010:;
      if(nat<mat) goto m675;
      if(nat==mat) goto m676;
      goto m677;
m675:;
      if(isd!=0) goto m5621;
      for(j=1;j<=kcel;j++)
        cdel[j]=rezc[j];
      for(j=1;j<=kdro;j++)
        ddel[j]=rezdr[j];
      rezc[1]=0;
      kcdel=kcel;
      kdDEL=kdRO;
      if(nat==0) goto m1231;
m5621:;
      if(isd==0) goto m8721;
      ddel[isd]=rezc[1];
      rezc[1]=0;
      kdDEL=isd;
m8721:;
      if(nat==0) goto m1231;
      for(j=1;j<=nat;j++)
      {
        if(a[j]!=0) goto m9782;
      }
      goto m1231;
m9782:;
      nat=nat+1;
      isd=isd+1;

      if(isd>krazr) goto m1231;

      a[nat]=0;
      for(i=1;i<=399;i++)
      {
        ctot[i]=0;

```

```

        cdtot[i]=0;
    }
    kctot=1;
    kdto=1;
    goto m2010;
m676:;
    for(j=1;j<=nat;j++)
    {
        if(a[j]>b[j]) goto m677;
        if(a[j]<b[j]) goto m675;
    }
m677:;
    wich(nat,kad,a,ad,mat,kbd,b,bd,&kc,&kcd,c,cd,krazr);
    sl(kctot,kdto,ctot,cdtot,kedi,kedid,ce,ced,
&kcel,&kdro,rezc,rezdr,krazr);
    kctot=kcel;
    kdto=kdro;
    for(i=1;i<=kcel;i++)
    ctot[i]=rezc[i];
    for(i=1;i<=kdro;i++)
    cdtot[i]=rezdr[i];
    nat=kc;
    kad=kcd;
    if(nat==0) a[1]=0;
    for(j=1;j<=nat;j++)
    a[j]=c[j];
    for(j=1;j<=kad;j++)
    ad[j]=cd[j];
    goto m2010;
m1231:;
    if(kcdel!=0) goto m971;
    kcdel=1;
    cdel[1]=0;
m971:;
    if(kddel!=0) goto m972;
    kddel=1;
    ddel[1]=0;
m972:;
    if(kddel>krazr) kddel=krazr;
    *pkcdel=kcdel;
    *pkddel=kddel;
    return;
}

```

**Набор dlmnkbc.h(определение номеров пунктов меню)**

```
#define K_1 102
#define K_3 103
#define K_4 104
#define K_5 105
#define K_6 106
#define K_7 107
#define K_8 108
#define P_1 109
#define P_2 110
#define P_3 111
#define P_4 112
#define P_5 113
#define P_6 114
#define P_7 115
#define P_8 116
#define K_2 101
#define ID_OBRAZ 200
#define ID_PORIW 201
#define ID_RUN 202
```

**Набор dlmnkbc.rc(файл ресурсов)**

```
#include "dlmnkbc.h"
#include <windows.h>
MYMENU MENU
{
    MENUITEM "run-F4", ID_RUN
}
MYMENU ACCELERATORS
{
    VK_F4, ID_RUN, VIRTKEY
}
MYDB DIALOG 37, 27, 239, 196
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "DIALOG DLJ WWOODA"
FONT 10, "Courier New Cyr"
{
    AUTOCHECKBOX "raspredelenie", P_1, 100, 11, 50, 14
    AUTOCHECKBOX "grafic", P_3, 100, 39, 50, 14
    PUSHBUTTON "OK", IDOK, 25, 120, 50, 14
    PUSHBUTTON "WIXOD", IDCANCEL, 105, 120, 50, 14
}
```

**Набор sleddlc(исходные данные для программы в BORLAND C++)**

(проба длинная мнк)

```
4 5
2
2 3
1 1
1
20 20
1 0 1 7 4
3 1 3 1 2
7 2 9 3 2
13 3 1 2 7
```

**Набор dlmnkbc(выходные данные для программы в BORLAND C++)**

(проба длинная мнк)

i=1 j=1

1 1 1

целая часть

1

дробная часть

0

i=1 j=2

1 1 1

целая часть

0

дробная часть

0

i=1 j=3

1 1 1

целая часть

1

дробная часть

0

i=1 j=4

1 1 1

целая часть

7

дробная часть

0

i=1 j=5

1 1 1

целая часть

4

дробная часть

0

i=2 j=1

1 1 1

целая часть

3

дробная часть  
0

$i=2$   $j=2$

1 1 1

целая часть  
1

дробная часть  
0

$i=2$   $j=3$

1 1 1

целая часть  
3

дробная часть  
0

$i=2$   $j=4$

1 1 1

целая часть  
1

дробная часть  
0

$i=2$   $j=5$

1 1 1

целая часть  
2

дробная часть  
0

$i=3$   $j=1$

1 1 1

целая часть  
7

дробная часть  
0

$i=3$   $j=2$

1 1 1

целая часть  
2

дробная часть  
0

i=3 j=3

1 1 1

целая часть

9

дробная часть

0

i=3 j=4

1 1 1

целая часть

3

дробная часть

0

i=3 j=5

1 1 1

целая часть

2

дробная часть

0

i=4 j=1

1 2 1

целая часть

13

дробная часть

0

i=4 j=2

1 1 1

целая часть

3

дробная часть

0

i=4 j=3

1 1 1

целая часть

1

дробная часть

0

i=4 j=4

1 1 1

целая часть

2

дробная часть

0

i=4 j=5

1 1 1

целая часть

7

дробная часть

0

степень полинома=1

зависимый параметр-1

(p1)

номера аргументов

2 3

оптимальная степень=1

Модель для зависимого параметра

аргумент-2

(p2)

номер аргумента-2 степень=1

знак=1

целая часть коэффициента

4

дробная часть коэффициента

14563106796116504856

аргумент-3

(p3)

номер аргумента-3 степень=1

знак=-1

целая часть коэффициента

0

дробная часть коэффициента

24271844660194174735

свободный член модели

знак свободного члена=1

целая часть свободного члена

0

дробная часть свободного члена  
63106796116504854350  
средняя абсолютная ошибка

целая часть ошибки  
0  
дробная часть ошибки  
52427184466019417439

номер аргумента=2

(p2)  
целая часть вклада  
0  
дробная часть вклада  
87980769230769230778

номер аргумента=3

(p3)  
целая часть вклада  
0  
дробная часть вклада  
12019230769230769221





1 1 1

целая часть

4

дробная часть

0

i=3 j=1

1 1 1

целая часть

2

дробная часть

0

i=3 j=2

-1 1 1

целая часть

1

дробная часть

0

i=3 j=3

1 1 1

целая часть

7

дробная часть

0

i=4 j=1

1 1 1

целая часть

3

дробная часть

0

i=4 j=2

1 1 1

целая часть

2

дробная часть

0

i=4 j=3

1 2 1

целая часть



0  
дробная часть вклада  
60000000000000000000

номер аргумента=2

(р 2)  
целая часть вклада  
0  
дробная часть вклада  
39999999999999999999

**ПРИЛОЖЕНИЕ-3(программа по реализации метода наименьших квадратов в длинной арифметике с распараллеливанием прямого хода в методе Гаусса,исходные и выходные данные)**

**Набор dlmnkmpr.c**

```

#include"mpi.h"
#include<stdio.h>
#include<math.h>
#include<alloc.h>
//с-метод наименьших квадратов в длинной арифметике
// при работе с UNIX уточнить работу с файлами и
//динамическое распределение памяти!
// здесь работа с файлами в С++ В WINDOWS!
FILE *fp4,*fp6,*fp5,*fp7,*fp9;
int main(int argc, char *argv[ ])
{
extern FILE *fp4,*fp6,*fp5,*fp7,*fp9;

extern void mnk(int argc, char *argv[ ],int n,int karg,int izn,
int kcn,int kdn,int *icn,int *idn,int *izx,int *kcx,int *kdx,
int *icx,int *idx,
int *izy,int *kcy,int *kdy,int *icy,int *idy,int k,
int *izak,int *kcak,int *kdak,
int *icak,int *idak,int *pigau,int krazr,int k50);

extern void ymnzn(int izw,int izr,int *pizrez,
int kcw,int kdw,int *icw,int *idw,
int kcr,int kdr,int *icr,int *idr,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern void wiczn(int izr,int izrl,int *pizrez,
int kcr,int kdr,int *icr,int *idr,

int kcl,int kdl,int *icl,int *idl,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern void slzn(int izr,int izrez,int *pizrl,
int kcr,int kdr,int *icr,int *idr,
int kcrez,int kdrez,int *icrez,int *idrez,
int *pkcl,int *pkdl,int *icl,int *idl,int krazr);

extern void delzn(int izr,int izn,int *pizrez,
int kcr,int kdr,int *icr,int *idr,
int kcn,int kdn,int *icn,int *idn,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

//с-метод наименьших квадратов в длинной арифметике
char anaz[101][61];int mnarg[21];char zgl[81];
/* integer*2 izrab(50,5),kcrab(50,20),kdrab(50,20),
*icrab(50,20,20),idrab(50,20,20),izy(50),kcy(50),kdy(50),

*icy(50,20),idy(50,20),izx(50,10),kcx(50,10),kdx(50,10),
*icx(50,10,20),idx(50,10,20),izakm(3,31),izak(31),kcakm(3,31),
*kcak(31),kdakm(3,31),kdak(31),icakm(3,31,20),icak(31,20),

```

```

*idakm(3,31,20),idak(31,20)
integer*2 izdelm(3),kcdelm(3),kddelm(3),icdelm(3,20),
*iddelm(3,20),izym(50),kcym(50),kdym(50),icym(50,20),idym(50,20), */
int izrab[10300], kcrab[10300], kdrab[10300],iz2,
icrab[215000], idrab[215000], izy[101], kcy[101], kdy[101],

icy[2200], idy[2200], izx[2200], kcx[2200],
kdx[2200],*pigau,*pizrez,*pkcrez,*pkdrez,
*pizrl,*pkcl,*pkdl,

icx[45000], idx[45000], izakm[510], izak[101], kcakm[510],
kcak[101], kdakm[510], kdak[101], icakm[11000], icak[2200],
idakm[11000], idak[2200];
int izdelm[5], kcdelm[5], kddelm[5], icdelm[110],
iddelm[110], izym[101], kcym[101], kdym[101], icym[2200],
idym[2200],k31,kakm,
icw[400],idw[400],icr[400],idr[400],icrez[400],idrez[400],
icl[400],
idl[400],icn[400],idn[400],icxmi[400],idxmi[400],icdelt[400],
iddelt[400], izwkl[21], kcwkl[21], kdwkl[21], icwkl[500], idwkl[500];
//izwkl[11],kcwkl[11],kdwkl[11],icwkl[11][20],
// idwkl[10][21]
int icsy[400],idsy[400],icwtot[400],idwtot[400];
double rwwod;long int icrww,idrww,nt;
int knaz,i,j,n,m,l,ios,izn,kcn,kdn,lt,kcel,kdel,k,kna,karg,
inay,ikoy,kmax,krazr,mis,ipy,jl,ila,iarg,narg,kml,k1,in,ik,
isp,ist,izw,kcw,kdw,izr,kcr,kdr,istep,isw,izrez,izr1,kcl,kdl,kcrez,
kdrez,

igau,izxmi,kcxmi,kdxmi,imin,kdro,kprod,iprod,ipech,km,kr,iz1,iz,
izsy,kcsy,kdsy,k50,izdelt,kcdelt,kddelt,izwtot,kcwtot,kdwtot;
// k50=10;
kakm=k31=100;
fp6=fopen("dlmnkbc","w");

/*
izrab =(int *)farmalloc(10300*4);
if(izrab ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izrab \n");
return 1;
}
kcrab =(int *)farmalloc(10300*4);
if(kcrab ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcrab \n");
return 1;
}
kdrab =(int *)farmalloc(10300*4);
if(kdrab ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdrab \n");
return 1;
}
icrab =(int *)farmalloc(215000*4);
if(icrab ==NULL)
{

fprintf(fp6,"NO MEMORY FOR icrab \n");
return 1;
}
idrab=(int *)farmalloc(215000*4);
if(idrab ==NULL)

```

```

{
fprintf(fp6,"NO MEMORY FOR idrab \n");
return 1;
}
izy =(int *)farmalloc(101*4);
if(izy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izy \n");
return 1;
}
kcy =(int *)farmalloc(101*4);
if(kcy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcy \n");
return 1;
}
kdy =(int *)farmalloc(101*4);
if(kdy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdy \n");
return 1;
}
icy =(int *)farmalloc(2200*4);
if(icy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icy \n");
return 1;
}
idy =(int *)farmalloc(2200*4);
if(idy ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idy \n");
return 1;
}
izx =(int *)farmalloc(2200*4);
if(izx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izx \n");
return 1;
}
kcx =(int *)farmalloc(2200*4);
if(kcx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcx \n");
return 1;
}
kdx =(int *)farmalloc(2200*4);
if(kdx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdx \n");
return 1;
}
icx =(int *)farmalloc(45000*4);
if(icx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icx \n");
return 1;
}
idx =(int *)farmalloc(45000*4);
if(idx ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idx \n");
return 1;
}

```

```

}
izakm =(int *)farmalloc(510*4);
if(izakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izakm \n");
return 1;
}
izak =(int *)farmalloc(101*4);
if(izak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izak \n");
return 1;
}
kcakm =(int *)farmalloc(510*4);
if(kcakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcakm \n");
return 1;
}
kcak =(int *)farmalloc(101*4);

if(kcak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcak \n");
return 1;
}

kdakm =(int *)farmalloc(510*4);
if(kdakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdakm \n");
return 1;
}
kdak =(int *)farmalloc(101*4);
if(kdak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdak \n");
return 1;
}
icakm =(int *)farmalloc(11000*4);
if(icakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icakm \n");
return 1;
}
icak =(int *)farmalloc(2200*4);
if(icak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icak \n");
return 1;
}
idakm =(int *)farmalloc(11000*4);
if(idakm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idakm \n");
return 1;
}
idak =(int *)farmalloc(2200*4);
if(idak ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idak \n");
return 1;
}
}

```

```

izdelm =(int *)farmalloc(5*4);
if(izdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izdelm \n");
return 1;
}
kcdelm =(int *)farmalloc(5*4);
if(kcdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcdelm \n");
return 1;
}

kddelm =(int *)farmalloc(5*4);
if(kddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kddelm \n");
return 1;
}
icdelm =(int *)farmalloc(110*4);
if(icdelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icdelm \n");
return 1;
}

iddelm =(int *)farmalloc(110*4);
if(iddelm ==NULL)
{
fprintf(fp6,"NO MEMORY FOR iddelm \n");
return 1;
}
izym =(int *)farmalloc(101*4);
if(izym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR izym \n");
return 1;
}
kcym =(int *)farmalloc(101*4);

if(kcym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kcym \n");
return 1;
}
kdym =(int *)farmalloc(101*4);
if(kdym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR kdym \n");
return 1;
}
icym =(int *)farmalloc(2200*4);

if(icym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR icym \n");
return 1;
}
idym =(int *)farmalloc(2200*4);
if(idym ==NULL)
{
fprintf(fp6,"NO MEMORY FOR idym \n");
return 1;
}

```

```

    }
    izwkl =(int *)farmalloc(21*4);
    if(izwkl ==NULL)
    {

        fprintf(fp6,"NO MEMORY FOR izwkl \n");
        return 1;
    }
    kcwkl =(int *)farmalloc(21*4);
    if(kcwkl ==NULL)
    {

        fprintf(fp6,"NO MEMORY FOR kcwkl \n");
        return 1;
    }
    kdwkl =(int *)farmalloc(21*4);
    if(kdwkl ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR kdwkl \n");
        return 1;
    }
    icwkl =(int *)farmalloc(500*4);
    if(icwkl ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR icwkl \n");
        return 1;
    }
    idwkl =(int *)farmalloc(500*4);
    if(idwkl ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR idwkl \n");
        return 1;
    }

}

*/
// fprintf(fp6,"\nmemory\n");
fp4=fopen("sleddlc","r");
fp7=fopen("name","r");
fp5=fopen("prowcdl2","w");
// open(4,file='sleddl',status='old')
// open(6,file='dlmnk2',status='Unknown')
// open(5,file='prowfdl2',status='Unknown')
// open(7,file='name',status='old')
//c open(9,file='prompr2',status='Unknown')
// read(7,*) knaz
fscanf(fp7,"%d",&knaz);

if(knaz>100) goto m1762;
for(i=1;i<=knaz;i++)
{
    for(j=1;j<=60;j++)
    anaz[i][j]=' ';
    for(j=1;j<=60;j++)
    {
        fscanf(fp7,"%c",&anaz[i][j]);
// fprintf(fp6,"%c",anaz[i][j]);
    }
// fprintf(fp6,"\n");
}
// return;
for(j=1;j<=80;j++)
{

```

```

fscanf(fp4,"%c",&zgl[j]);
fprintf(fp6,"%c",zgl[j]);
}
fprintf(fp6,"\n");
fscanf(fp4,"%d%d",&n,&m);
// printf("\nВведите количество параметров-аргументов<=10\n");
fscanf(fp4,"%d",&karg);
if(karg<=20) goto arg1;
fprintf(fp6,"\nколичество параметров-аргументов должно быть<=20!\n");
return 1;
arg1:;
// printf("введите через пробел номера аргументов\n");
for(i=1;i<=karg;i++)
fscanf(fp4,"%d",&mnarg[i]);
// printf("Введите через пробел начальный и конечный номер\n");
// printf("зависимой переменной\n");
fscanf(fp4,"%d%d",&inau,&ikoу);
// printf("\nвведите степень полинома<=3\n");
fscanf(fp4,"%d",&kmax);
if(kmax<=4) goto poll1;
fprintf(fp6,"степень полинома должна быть<=4\n");
return 1;
poll1:;
// printf("\nвведите количество разрядов для арифметики<=20\n");
fscanf(fp4,"%d%d",&krazr,&k50);

// fprintf(fp6,"\n n=%d m=%d\n",n,m);

// return;
if(n<=100) goto m1443;
fprintf(fp6,"число строчек-наблюдений должно быть<=100\n");
return 1;
m1443:;
if(m<=100) goto m1444;
m1762:;
fprintf(fp6,"число параметров-столбиков должно быть<=100\n");
return 1;
m1444:;
nt=n;
l=0;
m11601:;
ios=nt-nt/10*10;
l=l+1;
idl[l]=ios;
nt=nt/10;
if(nt>0) goto m11601;
izn=1;
kcn=1;
kdn=1;
lt=1;
for(l=1;l<=kcn;l++)
{
icn[l]=idl[lt];
lt=lt-1;
}
idn[l]=0;
//с-ввод исходной матрицы в длинном формате
for(i=1;i<=n;i++)
{
for(j=1;j<=m;j++)
{
fprintf(fp6,"\ni=%d j=%d\n",i,j);
fscanf(fp4,"%lf",&rwwod);

```

```

        icrww=fabs(rwwod);
        idrww=fabs((fabs(rwwod)-icrww)*pow(10.,9));
        izr=1;
        if(rwwod>=0) goto pre1;
        izr=-1;
pre1:;
        nt=icrww;
        l=0;
mpre1:;
        ios=nt-nt/10*10;
        l=l+1;
        id1[l]=ios;
        nt=nt/10;
        if(nt>0) goto mpre1;
        izrab[(i-1)*m+j]=izr;
        kcrab[(i-1)*m+j]=l;
        kcel=l;
        lt=l;
        iz1=l;
        for(l=1;l<=iz1;l++)
        {
            icrab[((i-1)*m+j-1)*k50+1]=id1[lt];
            lt=lt-1;
        }

        nt=idrww;
        l=0;
mpre2:;
        ios=nt-nt/10*10;
        l=l+1;
        id1[l]=ios;
        nt=nt/10;
        if(nt>0) goto mpre2;
        if(l==9) goto ul;
        kcl=9-l;
        for(jl=1;jl<=kcl;jl++)
        {
            l=l+1;
            id1[l]=0;
        }

ul:;
        kdrab[(i-1)*m+j]=l;
        lt=l;
        iz1=l;
        kdel=l;
        for(l=1;l<=iz1;l++)
        {
            idrab[((i-1)*m+j-1)*k50+1]=id1[lt];
            lt=lt-1;
        }

// write(6,*) izrab(i,j),kcrab(i,j),kdrab(i,j)
// fprintf(fp6,"\n%d %d %d\n",izr,
//         kcel,kdel);
// return;
//     izrab[(i-1)*m+j]=izr;
//     kcrab[(i-1)*m+j]=kcel;
//     kdrab[(i-1)*m+j]=kdel;
//     for(l=1;l<=kcel;l++)

```

```

//      {
//      fscanf(fp4,"%d",&izr);

//      fprintf(fp6,"%d",izr);
//      icrab[((i-1)*m+j-1)*k50+1]=izr;
//      }
//      return;
//      fprintf(fp6,"\n");
//      for(l=1;l<=kdel;l++)
//      {
//      fscanf(fp4,"%d",&izr);
//      fprintf(fp6,"%d",izr);
//      idrab[((i-1)*m+j-1)*k50+1]=izr;
//      }
//      fprintf(fp6,"\n");
//      return;
//      read(4,11002) (idrab(i,j,l),l=1,kdel)
//      fprintf(fp6,"\nцелая часть\n");
//      for(l=1;l<=kcel;l++)
//      fprintf(fp6,"%d",icrab[((i-1)*m+j-1)*k50+1]);
//      fprintf(fp6,"\n");
//      return;
//      fprintf(fp6,"\ndробная часть\n");
//      for(l=1;l<=kdel;l++)
//      fprintf(fp6,"%d",idrab[((i-1)*m+j-1)*k50+1]);
//      fprintf(fp6,"\n");
//      return;
//      }
//      }
//1002      continue
//7775      continue
//      return;
//      kna=1;
//      k=kna;
//      fprintf(fp6,"\nstепень полинома=%d\n",kmax);
//      mis=m;
//      m=karg;
//      return;
//      do 2001 ipy=inay,ikoу
//      for(ipy=inay;ipy<=ikoу;ipy++)
//      {
//      k=kna;
//      write(6,2003) ipy,(anaz(ipy,jl),jl=1,60)
//      fprintf(fp6,"\nzависимый параметр-%d\n",ipy);
//      for(jl=1;jl<=60;jl++)
//      fprintf(fp6,"%c",anaz[ipy][jl]);
//      fprintf(fp6,"\nномера аргументов\n");

//      for(i=1;i<=karg;i++)
//      fprintf(fp6,"%d ",mnarg[i]);
//      fprintf(fp6,"\n");
m403:;
//с-формирование вектора у для длинной арифметики
//      do 2002 i=1,n
//      for(i=1;i<=n;i++)
//      {
//      izy[i]=izrab[(i-1)*mis+ipy];
//      kcy[i]=kcrab[(i-1)*mis+ipy];
//      kdy[i]=kdrab[(i-1)*mis+ipy];
//      kcel=kcy[i];
//      kdel=kdy[i];
//      fprintf(fp6,"\n izy=%d kcy=%d kdy=%d\n",izy[i],kcel,kdel);
//      fprintf(fp6,"целая часть в главной\n");

```

```

        for(l=1;l<=kcel;l++)
        {
            icy[(i-1)*k50+1]=icrab[((i-1)*mis+ipy-1)*k50+1];
//          fprintf(fp6,"%d",icy[(i-1)*k50+1]);
        }
//          fprintf(fp6,"\ndробная часть в главной\n");
        for(l=1;l<=kdel;l++)
        {
            idy[(i-1)*k50+1]=idrab[((i-1)*mis+ipy-1)*k50+1];
//          fprintf(fp6,"%d",idy[(i-1)*k50+1]);
        }
    }
//c   y(i)=rab(i,ipy)
//2002   continue
//   printf("\n posle y\n");
//c-формирование матрицы x аргументов для длинной арифметики
    ila=0;
    for(iarg=1;iarg<=karg;iarg++)
    {

        narg=mnarg[iarg];
        ila=ila+1;
//      do 9002 i=1,n
        for(i=1;i<=n;i++)
        {
            izx[(i-1)*m+ila]=izrab[(i-1)*mis+narg];
            kcx[(i-1)*m+ila]=kcrab[(i-1)*mis+narg];
            kdx[(i-1)*m+ila]=kdrab[(i-1)*mis+narg];
            kcel=kcx[(i-1)*m+ila];
            kdel=kdx[(i-1)*m+ila];
            for(l=1;l<=kcel;l++)
            icx[((i-1)*m+ila-1)*k50+1]=icrab[((i-1)*mis+narg-1)*k50+1];

            for(l=1;l<=kdel;l++)
            idx[((i-1)*m+ila-1)*k50+1]=idrab[((i-1)*mis+narg-1)*k50+1];

//c   x(i,ila)=rab(i,narg)
//9002   continue
        } // end i
//9001   continue
    } //end iarg
//   printf("\npered mnk\n");
//   return;

    mnk(argc,argv,n,karg,izn,kcn,kdn,icn,idn,izx,kcx,kdx,icx,idx,
        izy,kcy,kdy,icy,idy,k,izak,kcak,kdak,icak,idak,&igau,krazr,k50);
//   fprintf(fp6,"\nposle mnk\n");
//   if(k==2) return;
//   printf("\n posle mnk\n");
//   return;
//   printf("\n igau=%d\n",igau);
//   fprintf(fp6,"posle mnk igau=%d\n",igau);
//   if(igau==0) goto m401;

    km1=k*m+1;
//   kakm=100;
//   do 8 i=1,km1
    for(i=1;i<=km1;i++)
    {
        izakm[(k-1)*kakm+i]=izak[i];
        kcakm[(k-1)*kakm+i]=kcak[i];
        kdakm[(k-1)*kakm+i]=kdak[i];
    }

```

```

kcel=kcak[i];
kdel=kdak[i];
for(l=1;l<=kcel;l++)
icakm[ ((k-1)*kakm+i-1)*k50+1]=icak[ (i-1)*k50+1];
for(l=1;l<=kdel;l++)
idakm[ ((k-1)*kakm+i-1)*k50+1]=idak[ (i-1)*k50+1];
}
//      printf("\n do 8\n");
//с-расчет модельных значений ym(i)
k1=k+1;
izdelm[k]=1;
kcdelm[k]=1;
kddelm[k]=1;
icdelm[ (k-1)*k50+1]=0;
iddelm[ (k-1)*k50+1]=0;

//c  delm(k)=0
//  do 54 i=1,n
for(i=1;i<=n;i++)
{
izym[i]=izak[km1];
kcym[i]=kcak[km1];
kdym[i]=kdak[km1];
kcel=kcak[km1];
kdel=kdak[km1];
for(l=1;l<=kcel;l++)
icym[ (i-1)*k50+1]=icak[ (km1-1)*k50+1];
for(l=1;l<=kdel;l++)
idym[ (i-1)*k50+1]=idak[ (km1-1)*k50+1];

//c  ym(i)=ak(km1)
//  do 540 j=1,m
for(j=1;j<=m;j++)
{
in=(j-1)*(k)+1;
ik=in+k-1;
isp=1;
//  do 55 ist=in,ik
for(ist=in;ist<=ik;ist++)
{
//c  ym(i)=ym(i)+ak(ist)*x(i,j)**(k-(isp-1))
izw=1;
kcw=1;
kdw=1;
icw[1]=1;
idw[1]=0;
izr=izx[ (i-1)*m+j];
kcr=kcx[ (i-1)*m+j];
kdr=kdx[ (i-1)*m+j];
for(l=1;l<=kcr;l++)
icr[l]=icx[ ((i-1)*m+j-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idx[ ((i-1)*m+j-1)*k50+1];
istep=k-(isp-1);
if(istep==0) goto m11103;

//  do 11104 isw=1,istep
for(isw=1;isw<=istep;isw++)
{
ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
izw=izrez;

```

```

    kcw=kcrez;
    kdw=kdrez;
    for (l=1;l<=kcw;l++)
    icw[l]=icrez[l];
    for (l=1;l<=kdw;l++)
    idw[l]=idrez[l];
    }
m11103:;
    izr=izak[ist];
    kcr=kcak[ist];
    kdr=kdak[ist];
    for (l=1;l<=kcr;l++)

    icr[l]=icak[(ist-1)*k50+1];
    for (l=1;l<=kdr;l++)
    idr[l]=idak[(ist-1)*k50+1];
    ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
    kcr,kdr,icr,idr,
    &kcrez,&kdrez,icrez,idrez,krazr);
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for (l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for (l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];
    slzn(izr,izrez,&izr1,kcr,kdr,icr,idr,
    kcrez,kdrez,icrez,idrez,
    &kc1,&kd1,ic1,id1,krazr);
    izym[i]=izr1;
    kcym[i]=kc1;
    kdym[i]=kd1;
    for (l=1;l<=kc1;l++)
    icym[(i-1)*k50+1]=ic1[l];
    for (l=1;l<=kd1;l++)
    idym[(i-1)*k50+1]=id1[l];
    isp=isp+1;
//55 continue
    }

//540 continue
    }
// printf("\n 540\n");
//c pause '540'
//c delm(k)=delm(k)+dabs(ym(i)-y(i))
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for (l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for (l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];

    izr1=izy[i];
    kc1=ky[i];
    kd1=ky[i];
    for (l=1;l<=kc1;l++)
    ic1[l]=icy[(i-1)*k50+1];
    for (l=1;l<=kd1;l++)
    id1[l]=idy[(i-1)*k50+1];
    wiczn(izr,izr1,&izrez,kcr,kdr,icr,idr,
    kc1,kd1,ic1,id1,

```

```

&kcrez, &kdrez, icrez, idrez, krazr);
  izrez=izrez*izrez;
  izr=izdelm[k];
  kcr=kcdelm[k];
  kdr=kddelm[k];
  for (l=1;l<=kcr;l++)
    icr[l]=icdelm[(k-1)*k50+1];
  for (l=1;l<=kdr;l++)
    idr[l]=iddelm[(k-1)*k50+1];
  slzn(izr, izrez, &izr1, kcr, kdr, icr, idr,
kcrez, kdrez, icrez, idrez,
&kcl, &kdl, icl, idl, krazr);
  izdelm[k]=izr1;
  kcdelm[k]=kcl;
  kddelm[k]=kdl;
  for (l=1;l<=kcl;l++)
    icdelm[(k-1)*k50+1]=icl[l];

  for (l=1;l<=kdl;l++)
    iddelm[(k-1)*k50+1]=idl[l];
//54 continue

}
//      printf("\n 54\n");
//c  pause '54'
//c  delm(k)=delm(k)/n
  izr=izdelm[k];
  kcr=kcdelm[k];
  kdr=kddelm[k];
  for (l=1;l<=kcr;l++)
    icr[l]=icdelm[(k-1)*k50+1];
  for (l=1;l<=kdr;l++)
    idr[l]=iddelm[(k-1)*k50+1];
  delzn(izr, izn, &izrez, kcr, kdr, icr, idr,
kcn, kdn, icn, idn,
&kcrez, &kdrez, icrez, idrez, krazr);
  izdelm[k]=izrez;
  kcdelm[k]=kcrez;
  kddelm[k]=kdrez;
  for (l=1;l<=kcrez;l++)
    icdelm[(k-1)*k50+1]=icrez[l];
  for (l=1;l<=kdrez;l++)
    iddelm[(k-1)*k50+1]=idrez[l];
  k=k+1;
  if(k<=kmax) goto m403;
//401 continue
m401:;
//      printf("\n m401\n");
//c  pause '401'
  if((igau==0)&&(k==1)) goto m9988;
  if(k<kmax) kmax=k-1;
  if((k==kmax)&&(igau==0)) kmax=k-1;
  izxmi=izdelm[kna];
  kcxmi=kcdelm[kna];
  kdxmi=kddelm[kna];
  for (l=1;l<=kcxmi;l++)
    icxmi[l]=icdelm[(kna-1)*k50+1];
  for (l=1;l<=kdxmi;l++)
    idxmi[l]=iddelm[(kna-1)*k50+1];
//c  xmin=delm(kna)
  imin=kna;

```

```

// do 405 i=kna,kmax
for (i=kna;i<=kmax;i++)
{
    kcel=kcdelm[i];
    kdel=kddelm[i];
    if(kcel>kcxmi) goto m405;
    if(kcel<kcxmi) goto m12101;
    for (l=1;l<=kcel;l++)
    icr[l]=icdelm[(i-1)*k50+1];
    for (l=1;l<=kdel;l++)
    idr[l]=iddelm[(i-1)*k50+1];
    for (l=1;l<=kcel;l++)
    {
        if(icr[l]>icxmi[l]) goto m405;
        if(icr[l]<icxmi[l]) goto m12101;
    }
    kdro=kdel;
    if(kdel>kdxmi) kdro=kdxmi;
    for (l=1;l<=kdro;l++)
    {
        if(idr[l]>idxmi[l]) goto m405;
        if(idr[l]<idxmi[l]) goto m12101;
    }

    if(kdel>=kdxmi) goto m405;

//c if(kdel.eq.kdxmi) goto 405

//c if(delm(i).ge.xmin) goto 405
m12101;;
//c xmin=delm(i)
    izxmi=izdelm[i];
    kcxmi=kcdelm[i];
    kdxmi=kddelm[i];
    for (l=1;l<=kcxmi;l++)
    icxmi[l]=icdelm[(i-1)*k50+1];

    for (l=1;l<=kdxmi;l++)
    idxmi[l]=iddelm[(i-1)*k50+1];
    imin=i;
m405;;
}

//с-проверить необходимость метки 406
//406 continue
//с pause '406'
//с-расчет модельных значений um(i) по min ошибке
    fprintf(fp6, "\ноптимальная степень=%d\n", imin);

    fprintf(fp6, "\nМодель для зависимого параметра\n");
    kprod=7;
    iprod=0;
    ipech=0;
// do 4001 iarg=1,karg
for (iarg=1;iarg<=karg;iarg++)
{
    narg=mnarg[iarg];
    fprintf(fp6, "\nаргумент-%d\n", narg);
    for (jl=1;jl<=60;jl++)
    fprintf(fp6, "%c", anaz[narg][jl]);
}

```

```

kml=imin*m+1;
km=imin*m;
kr=imin;

in=(iarg-1)*imin+1;
ik=in+imin-1;

// do 876 j=in,ik
for(j=in;j<=ik;j++)
{

// write(6,12451) narg,kr
fprintf(fp6,"\nномер аргумента-%d степень=%d\n",narg,kr);
// k31=100;
iz1=izakm[(imin-1)*k31+j];
kcel=kcakm[(imin-1)*k31+j];
kdel=kdakm[(imin-1)*k31+j];
for(l=1;l<=kcel;l++)
icr[l]=icakm[((imin-1)*k31+j-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[((imin-1)*k31+j-1)*k50+1];

fprintf(fp6,"\nзнак=%d\n",iz1);

fprintf(fp6,"\nцелая часть коэффициента\n");
for(l=1;l<=kcel;l++)
fprintf(fp6,"%1d",icr[l]);
fprintf(fp6,"\n");
fprintf(fp6,"\ndробная часть коэффициента\n");
for(l=1;l<=kdel;l++)
fprintf(fp6,"%1d",idr[l]);
fprintf(fp6,"\n");
//c write(9,50211) akm(imin,j),narg,kr
//c write(6,50211) akm(imin,j),narg,kr

kr=kr-1;
m876:;
}
m4001:;
}
//c write(6,14007) akm(imin,kml)
//c write(9,14007) akm(imin,kml)
//c write(*,14007) akm(imin,kml)
//c pause

iz=izakm[(imin-1)*k31+kml];
kcel=kcakm[(imin-1)*k31+kml];
kdel=kdakm[(imin-1)*k31+kml];
for(l=1;l<=kcel;l++)
icr[l]=icakm[((imin-1)*k31+kml-1)*k50+1];

for(l=1;l<=kdel;l++)
idr[l]=idakm[((imin-1)*k31+kml-1)*k50+1];

fprintf(fp6,"\nсвободный член модели\n");

fprintf(fp6,"\nзнак свободного члена=%d\n",iz);

fprintf(fp6,"\nцелая часть свободного члена\n");
for(l=1;l<=kcel;l++)

```



```

icw[l]=1;
idw[l]=0;
izr=izx[(i-1)*m+j];
kcr=kcx[(i-1)*m+j];
kdr=kdx[(i-1)*m+j];
for(l=1;l<=kcr;l++)

icr[l]=icx[((i-1)*m+j-1)*k50+1];

for(l=1;l<=kdr;l++)
idr[l]=idx[((i-1)*m+j-1)*k50+1];

// if(istep==0) goto m31103;
do 31104 isw=1,istep
for(isw=1;isw<=istep;isw++)
{
ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
izw=izrez;
kcw=kcrez;
kdw=kdrez;
for(l=1;l<=kcw;l++)
icw[l]=icrez[l];
for(l=1;l<=kdw;l++)
idw[l]=idrez[l];

//31104 continue
}
m31103:;
izr=izakm[(imin-1)*k31+ist];
kcr=kcakm[(imin-1)*k31+ist];

kdr=kdakm[(imin-1)*k31+ist];
for(l=1;l<=kcr;l++)
icr[l]=icakm[((imin-1)*k31+ist-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idakm[((imin-1)*k31+ist-1)*k50+1];

ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
iz1=izym[i];
kcym[kcr]=kcym[i];
kdym[kdr]=kdym[i];
for(l=1;l<=kcr;l++)
icr[l]=icym[(i-1)*k50+1];

for(l=1;l<=kdr;l++)
idr[l]=idym[(i-1)*k50+1];

slzn(iz1,izrez,&iz2,kcr,kdr,icr,idr,
kcrez,kdrez,icrez,idrez,
&kc1,&kdl,ic1,id1,krazr);
izym[i]=iz2;
kcym[i]=kc1;
kdym[i]=kdl;
for(l=1;l<=kc1;l++)
icym[(i-1)*k50+1]=ic1[l];
for(l=1;l<=kdl;l++)
idym[(i-1)*k50+1]=id1[l];

//c ym(i)=ym(i)+akm(imin,ist)*x(i,j)**(imin-(isp-1))

```

```

    iz1=izwkl[j];
    kcel=kcwkl[j];

    kdel=kdwkl[j];
    for(l=1;l<=kcel;l++)
    icr[l]=icwkl[(j-1)*k50+1];
    for(l=1;l<=kdel;l++)
    idr[l]=idwkl[(j-1)*k50+1];
    izrez=izrez*izrez;
    slzn(iz1,izrez,&iz2,kcel,kdel,icr,idr,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
//c   wkl(j)=wkl(j)+dabs(akm(imin,ist)*x(i,j)**(imin-(isp-1)))

    izwkl[j]=iz2;
    kcwkl[j]=kcl;
    kdwkl[j]=kdl;
    for(l=1;l<=kcl;l++)
    icwkl[(j-1)*k50+1]=icl[l];
    for(l=1;l<=kdl;l++)
    idwkl[(j-1)*k50+1]=idl[l];

    isp=isp+1;
//3355     continue
    }
//33540     continue

    }
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for(l=1;l<=kdr;l++)
    idr[l]=idym[(i-1)*k50+1];

    iz1=izakm[(imin-1)*k31+km1];
    kcl=kcakm[(imin-1)*k31+km1];
    kdl=kdakm[(imin-1)*k31+km1];
    for(l=1;l<=kcl;l++)
    icl[l]=icakm[((imin-1)*k31+km1-1)*k50+1];
    for(l=1;l<=kdl;l++)
    idl[l]=idakm[((imin-1)*k31+km1-1)*k50+1];
    wiczn(izr,iz1,&izrez,kcr,kdr,icr,idr,
kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
//c   sy=sy+ym(i)-akm(imin,km1)
    slzn(izsy,izrez,&iz2,kcsy,kdsy,icsy,idsy,
kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
    izsy=iz2;
    kcsy=kcl;
    kdsy=kdl;
    for(l=1;l<=kcl;l++)
    icsy[l]=icl[l];
    for(l=1;l<=kdsy;l++)
    idsy[l]=idl[l];
    izr=izym[i];
    kcr=kcym[i];
    kdr=kdym[i];
    for(l=1;l<=kcr;l++)
    icr[l]=icym[(i-1)*k50+1];
    for(l=1;l<=kdr;l++)

```

```

        idr[l]=idym[(i-1)*k50+1];

/*c  write(6,*) i
c    write(6,*) 'модельное y'
c    write(6,*) izr
c    write(6,*) 'целая часть'
c    write(6,87321) (icr(l),l=1,kcr)

87321 format(1x,60i1)
c    write(6,*) 'дробная часть'
c    write(6,87321) (idr(l),l=1,kdr) */
        iz1=izy[i];
        kc1=кcy[i];
        kd1=кdy[i];
        for(l=1;l<=kc1;l++)
            ic1[l]=icy[(i-1)*k50+1];
        for(l=1;l<=kd1;l++)
            id1[l]=idy[(i-1)*k50+1];
/*c  write(6,*) 'исходное y'

c    write(6,*) iz1

c    write(6,*) 'целая часть'
c    write(6,87321) (ic1(l),l=1,kc1)
c    write(6,*) 'дробная часть'
c    write(6,87321) (id1(l),l=1,kd1)      */
        wiczn(izr,iz1,&iz2,kcr,kdr,icr,idr,
        kc1,kd1,ic1,id1,
        &kcrez,&kdrez,icrez,idrez,krazr);
        izrez=iz2*iz2;
//c  delt=delt+dabs(ym(i)-y(i))
        slzn(izdelt,izrez,&iz2,kcdelt,kddelt,icdelt,iddelt,
        kcrez,kdrez,icrez,idrez,
        &kc1,&kd1,ic1,id1,krazr);
        izdelt=iz2;
        kcdelt=kc1;
        kddelt=kd1;
        for(l=1;l<=kc1;l++)
            icdelt[l]=ic1[l];
        for(l=1;l<=kd1;l++)
            iddelt[l]=id1[l];

//c345      format(1x,'x=',f8.4,' yi=',f15.5,' ym=',f15.5,' del=',f15.5)
m3354:;
    }
    izwtot=1;
    kcwtot=1;
    kdwtot=1;
    icwtot[1]=0;
    idwtot[1]=0;
//c  wkltot=0
//  do 5204 i=1,m
    for(i=1;i<=m;i++)
    {
        izr=1;
        kcr=kcwk1[i];
        kdr=kdwk1[i];
        for(l=1;l<=kcr;l++)
            icr[l]=icwk1[(i-1)*k50+1];
        for(l=1;l<=kdr;l++)
            idr[l]=idwk1[(i-1)*k50+1];
//c  wkltot=wkltot+dabs(wkl(i))
        slzn(izwtot,izr,&iz1,kcwtot,kdwtot,icwtot,idwtot,

```

```

kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
izwtot=izl;
kcwtot=kcrez;
kdwtot=kdrez;
for(l=1;l<=kcrez;l++)
icwtot[l]=icrez[l];
for(l=1;l<=kdrez;l++)
idwtot[l]=idrez[l];
//5204      continue
}
for(i=1;i<=m;i++)
{
izr=izwkl[i];
kcr=kcwkl[i];
kdr=kdwkl[i];
for(l=1;l<=kcr;l++)
icr[l]=icwkl[(i-1)*k50+1];
for(l=1;l<=kdr;l++)
idr[l]=idwkl[(i-1)*k50+1];
//c      wk1(i)=wk1(i)/wkltot
delzn(izr,izwtot,&izl,kcr,kdr,icr,idr,
kcwtot,kdwtot,icwtot,idwtot,
&kcl,&kdl,icl,idl,krazr);
izwkl[i]=izl;
kcwkl[i]=kcl;
kdwkl[i]=kdl;
for(l=1;l<=kcl;l++)
icwkl[(i-1)*k50+1]=icl[l];
for(l=1;l<=kdl;l++)
idwkl[(i-1)*k50+1]=idl[l];
//5203      continue
//5202      continue
}
//c      delt=delt/n

delzn(izdelt,izn,&izr,kcdelt,kddelt,icdelt,iddelt,

kcn,kdn,icn,idn,

&kcrez,&kdrez,icrez,idrez,krazr);
izdelt=izr;
kcdelt=kcrez;
kddelt=kdrez;
for(l=1;l<=kcrez;l++)
icdelt[l]=icrez[l];
for(l=1;l<=kdrez;l++)
iddelt[l]=idrez[l];

fprintf(fp6,"\nсредняя абсолютная ошибка\n");
fprintf(fp6,"\nцелая часть ошибки\n");
for(l=1;l<=kcdelt;l++)
fprintf(fp6,"%1d",icdelt[l]);
fprintf(fp6,"\ndробная часть ошибки\n");
for(l=1;l<=kddelt;l++)
fprintf(fp6,"%1d",iddelt[l]);

fprintf(fp6,"\n");
//      prob=' '
//      do 5404 i=1,m
for(i=1;i<=m;i++)

```

```

    {
    narg=mnarg[i];
// write(6,5405) narg, (anaz(narg,j),j=1,60)
    fprintf(fp6,"\nномер аргумента=%d\n",narg);
    for(j=1;j<=60;j++)
    fprintf(fp6,"%c",anaz[narg][j]);
    fprintf(fp6,"\nцелая часть вклада\n");
    kcel=kcwkl[i];
    for(l=1;l<=kcel;l++)
    fprintf(fp6,"%ld",icwkl[(i-1)*k50+l]);
    fprintf(fp6,"\ndробная часть вклада\n");
    kdel=kdwkl[i];
    for(l=1;l<=kdel;l++)
    fprintf(fp6,"%ld",idwkl[(i-1)*k50+l]);
    fprintf(fp6,"\n");

//5404      continue

    }
//891 format(/lx,'средняя абсолютная ошибка=',f36.18/)
m9988;;
m2001;;
    }

    fclose(fp4);
    fclose(fp6);
    fclose(fp5);
    fclose(fp7);
    return 1;
    }

//Набор mnk.c(модификация для распараллеливания)
#include<stdio.h>
#include<math.h>
#include<alloc.h>
// УТОЧНИТЬ РАБОТУ С ФАЙЛАМИ В UNIX,ЗДЕСЬ РАБОТА В C++ WINDOWS!

    void mnk(int argc, char *argv[ ],int n,int m,int izn,int kcn,
int kdn,int *icn,int *idn,int *izx,int *kcx,int *kdx,
int *icx,int *idx,
    int *izy,int *kcy,int *kdy,
int *icy,int *idy,int k,int *izak,int *kcak,int *kdak,
int *icak,int *idak,int *pigau,int krazr,int k50)
    {
extern FILE *fp6,*fp9;

    extern void ymnzn(int izw,int izr,int *pizrez,
int kcw,int kdw,int *icw,int *idw,
int kcr,int kdr,int *icr,int *idr,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

    extern void slzn(int izr,int izrez,int*pizl,
int kcr,int kdr,int *icr,int *idr,
int kcrez,int kdrez,int *icrez,int *idrez,
int *pkcl,int *pkdl,int *icl,int *idl,int krazr);

    extern void gaydl(int argc, char *argv[ ],int kg,int *izc,int *kcc,
int *kdc,
int *icc,int *idc,
int *izf,int *kcf,int *kdf,int *icf,int *idf,

```

```

int *izak,int *kcak,int *kdak,int *icak,
    int *idak,int *pigau,int krazr,int k50,int *mnper);

extern void prowg(int kg,int *izc,int *kcc,int *kdc,
    int *icc,int *idc,
int *izf,int *kcf,int *kdf,int *icf,int *idf,
    int *izak,int *kcak,int *kdak,int *icak,int *idak,
    int *pigau,int krazr,int k50);

//с-метод наименьших квадратов в длинной арифметике
    int izf[101], kcf[101], kdf[101], icf[2200], idf[2200],mnper[100],
        *pizrez,*pkcrez,*pizl,*pkcl,*pkdl,
        icw[400],idw[400],ilf,
//izx(50,10),kcx(50,10),kdx(50,10)
//    integer*2 icx(50,10,20),idx(50,10,20),
icrez[400],idrez[400],
//    *izy(50),kcy(50),kdy(50),icy(50,20),idy(50,20),
icr[400],idr[400],
    izc[10300], kcc[10300], kdc[10300], icc[215000], idc[215000],
    icwl[400],idwl[400],icl[400],idl[400];
//icn(400),idn(400)
//    integer*2 izak(31),kcak(31),kdak(31),icak(31,20),idak(31,20)
int s,p,jp,ip,igau,igur;
int k1,i,izw,kcw,kdw,izr,kcr,kdr,l,istep,izrez,kcrez,kdrez,lp,izl,
kcl,kdl,j,jlc,ilc,kzmat,lpr,izwl,kdwl,kcwl,kmat,istr,isto,inf,
iccl,jc1,kraz,kcel,kdel,kg;
kzmat=k*m+1;kmat=k*m+1;

/*
    izf =(int *)farmalloc(101*4);
    if(izf ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR izf \n");
        return;
    }
    kcf =(int *)farmalloc(101*4);
    if(kcf ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR kcf \n");
        return;
    }
    kdf =(int *)farmalloc(101*4);
    if(kdf ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR kdf \n");
        return;
    }
    icf =(int *)farmalloc(2200*4);
    if(icf ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR icf \n");
        return;
    }
    idf =(int *)farmalloc(2200*4);
    if(idf ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR idf \n");
        return;
    }
    izc =(int *)farmalloc(10300*4);

```

```

    if(izc ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR izc \n");
        return;
    }
    kcc =(int *)farmalloc(10300*4);
    if(kcc ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR kcc \n");
        return;
    }
    kdc =(int *)farmalloc(10300*4);
    if(kdc ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR kdc \n");
        return;
    }
    icc=(int *)farmalloc(215000*4);
    if(icc ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR icc \n");
        return;
    }
    idc =(int *)farmalloc(215000*4);

    if(idc ==NULL)
    {
        fprintf(fp6,"NO MEMORY FOR idc \n");
        return;
    }
*/

//

//с-с(50,50), матрица системы нормальных уравнений
//с-f(50), вектор свободных членов
//с-ak(50), коэффициенты полиномов по аргументам
//с-х(50,10), матрица аргументов
//с-у(50), массив значений функции
//с-ум(50), массив модельных значений функции
//с write(6,*) '          Метод наименьших квадратов'
/* с-определение матрицы с
с-і-номер наблюдения, строки
с-j-номер параметра-аргумента
с-l-номер коэффициента, степени

с-s-номер степени, р-номер параметра
с-m-количество параметров-аргументов
с-k, порядок полиномов
с-n, число значений функции */
//  fprintf(fp6, "\nпередача в mnk\n");

//  fprintf(fp6, "n=%d m=%d k50=%d krazr=%d k=%d\n", n, m,
//          k50, krazr, k);
//  printf("\n mnk  n=%d m=%d k50=%d krazr=%d\n", n, m, k50, krazr);
//  scanf("%d", &iz1);
//  fprintf(fp6, "\nматрица x\n");
//с do 34521 i=1, n
//  for(i=1; i<=n; i++)
//  {
//с do 34522 j=1, m

```

```

//   for(j=1;j<=m;j++)
//   {
//   fprintf(fp6,"\n i=%d j=%d\n", i,j);
//   fprintf(fp6,"\nцелая часть x\n");
//   izl=izx[(i-1)*m+j];
//   fprintf(fp6,"\nзнак=%d \n",izl);
//   kcel=kcx[(i-1)*m+j];
//   kdel=kdx[(i-1)*m+j];
//   for(l=1;l<=kcel;l++)
//   fprintf(fp6,"%d",icx[((i-1)*m+j-1)*k50+1]);

//   fprintf(fp6,"\nдробная часть x\n");
//   for(l=1;l<=kdel;l++)
//   fprintf(fp6,"%d",idx[((i-1)*m+j-1)*k50+1]);
//34522   continue
//34521   continue
//   }
//   }
//   fprintf(fp6,"\nвектор y\n");
//   do 34545 i=1,n
/*   for(i=1;i<=n;i++)
    {
        kcel=ky[i];
        kdel=ky[i];
        fprintf(fp6,"\n mnk знак=%d kcel=%d kdel=%d \n",
            izy[i],kcel,kdel);
//   printf(" \nmnk zn=%d kcel=%d kdel=%d\n",izy[i],kcel,kdel);
//   scanf("%d",&izl);
        fprintf(fp6,"\nцелая часть y\n");
//   printf("\nцелая часть\n");
        for(l=1;l<=kcel;l++)
        {
//   printf("%d",icy[(i-1)*k50+1]);
//   scanf("%d",&izl);
            fprintf(fp6,"%d",icy[(i-1)*k50+1]);
        }
//   printf("\nдробная часть\n");
        fprintf(fp6,"\n дробная часть y \n");
        for(l=1;l<=kdel;l++)
        {
//   printf("%d",idy[(i-1)*k50+1]);
//   scanf("%d",&izl);
            fprintf(fp6,"%d",idy[(i-1)*k50+1]);
        }
//   printf("\ni=%d\n",i);
//   scanf("%d",&izl);
//34545   continue
    }
//c   stop           */
    k1=k+1;
//c   pause 'mnk1'
//   do 2 p=1,m
//   printf(" mnk do2 \n");
//   scanf("%d",&igur);
//   return;
//   fprintf(fp6,"\nwid y\n");
    for(p=1;p<=m;p++)
    {

//   do 282 s=1,k
    for(s=1;s<=k;s++)
    {
//   fprintf(fp6,"\n p=%d s=%d\n",p,s);

```

```

    izf[(p-1)*k+s]=1;
    kcf[(p-1)*k+s]=1;
    kdf[(p-1)*k+s]=1;
    icf[((p-1)*k+s-1)*k50+1]=0;
    idf[((p-1)*k+s-1)*k50+1]=0;
//c f((p-1)*k+s)=0
// do 2002 i=1,n
    for(i=1;i<=n;i++)
    {
// fprintf(fp6,"\n i=%d\n",i);
    izw=1;
    kcw=1;
    kdw=1;
    icw[1]=1;
    idw[1]=0;
    izr=izx[(i-1)*m+p];
    kcr=kcx[(i-1)*m+p];
    kdr=kdx[(i-1)*m+p];
    for(l=1;l<=kcr;l++)
    icr[l]=icx[((i-1)*m+p-1)*k50+1];
    for(l=1;l<=kdr;l++)
    idr[l]=idx[((i-1)*m+p-1)*k50+1];
    istep=k-(s-1);
    if(istep==0) goto m32103;
// do 12104 l=1,istep
    for(l=1;l<=istep;l++)
    {

        ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
//c write(6,*) kcw,kdw,kcr,kdr,kcrez,kdrez
//c pause '2'
        izw=izrez;
        kcw=kcrez;
        kdw=kdrez;
        for(lp=1;lp<=kcrez;lp++)
        icw[lp]=icrez[lp];
        for(lp=1;lp<=kdrez;lp++)
        idw[lp]=idrez[lp];
//12104 continue
    }

m32103:;
// fprintf(fp6,"\nm32103\n");
//c pause '32103'
    izr=izy[i];
    kcr=kcy[i];
    kdr=kdy[i];
    for(l=1;l<=kcr;l++)
    icr[l]=icy[(i-1)*k50+1];
    for(l=1;l<=kdr;l++)
    idr[l]=idy[(i-1)*k50+1];
    ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
// fprintf(fp6,"\nposle ymnzn\n");
    ilf=(p-1)*k+s;
    izr=izf[ilf];
    kcr=kcf[ilf];
    kdr=kdf[ilf];
    for(l=1;l<=kcr;l++)
    icr[l]=icf[(ilf-1)*k50+1];

```

```

        for(l=1;l<=kdr;l++)
            idr[l]=idf[(ilf-1)*k50+1];
/* c pause '5 na'
c write(6,*) 'первое слагаемое'
c write(6,*) izr,kcr,kdr
c write(6,*) izr,kcr,kdr
c pause '5_1'
c write(6,*) izr

c write(6,49701) izr
49701 format(1x,'знак=',i3/)
c write(6,*) 'целая часть'
c write(6,*) (icr(l),l=1,kcr)
c pause '5_1_1'
c write(6,*) (icr(l),l=1,kcr)
c pause
49702 format(1x,50i1)
c write(6,*) 'дробная часть'
c write(6,*) (idr(l),l=1,kdr)
49703 format(1x,50i1)
c write(6,*) 'второе слагаемое'

c write(6,49701) izrez
c write(6,*) izrez,kcrez,kdrez
c pause '5_2'
c write(6,*) 'целая часть'
c write(6,*) (icrez(l),l=1,kcrez)
c write(6,*) 'дробная часть'
c write(6,*) (idrez(l),l=1,kdrez)
c pause '5' */
slzn(izr,izrez,&izl,kcr,kdr,icr,idr,

        kcrez,kdrez,icrez,idrez,
&kcl,&kdl,icl,idl,krazr);
// fprintf(fp6,"\nposle slzn\n");
//c pause '6'
//c f((p-1)*k+s)= f((p-1)*k+s)+y(i)*x(i,p)**(k-(s-1))
izf[ilf]=izl;
kcf[ilf]=kcl;
kdf[ilf]=kdl;
for(l=1;l<=kcl;l++)
icf[(ilf-1)*k50+1]=icl[l];
for(l=1;l<=kdl;l++)
idf[(ilf-1)*k50+1]=idl[l];
// fprintf(fp6,"\npered 2002\n");
//2002 continue
}
// fprintf(fp6,"\n 2002\n");
// scanf("%d",&igur);
//c pause '2002'
// do 212 j=1,m
for(j=1;j<=m;j++)
{

// do 211 l=1,k

for(l=1;l<=k;l++)
{
// fprintf(fp6," \nmnk j=%d l=%d\n",j,l);
ilc=(p-1)*k+s;
jlc=(j-1)*k+l;

```

```

//c  c((p-1)*k+s, (j-1)*k+1)=0
      kzmat=k*m+1;

      izc[(i1c-1)*kzmat+j1c]=1;
      kcc[(i1c-1)*kzmat+j1c]=1;
      kdc[(i1c-1)*kzmat+j1c]=1;
      icc[((i1c-1)*kzmat+j1c-1)*k50+1]=0;
      idc[((i1c-1)*kzmat+j1c-1)*k50+1]=0;
//
do 3 i=1,n
  for(i=1;i<=n;i++)
  {

//c  c((p-1)*k+s, (j-1)*k+1)= c((p-1)*k+s, (j-1)*k+1)+
//c    *x(i,j)**(k-(l-1))*x(i,p)**(k-(s-1))
      izw=1;
      kcw=1;
      kdw=1;
      icw[1]=1;
      idw[1]=0;
      istep=k-(l-1);
      izr=izx[(i-1)*m+j];
      kcr=kcx[(i-1)*m+j];
      kdr=kdx[(i-1)*m+j];
      for(lp=1;lp<=kcr;lp++)
        icr[lp]=icx[((i-1)*m+j-1)*k50+lp];
      for(lp=1;lp<=kdr;lp++)
        idr[lp]=idx[((i-1)*m+j-1)*k50+lp];
      if(istep==0) goto m16101;
//
do 16102 lpr=1,istep
  for(lpr=1;lpr<=istep;lpr++)
  {

//c  pause '7'
      ymnzn(izw,izr,&izl,kcw,kdw,icw,idw,
            kcr,kdr,icr,idr,
            &kcrez,&kdrez,icrez,idrez,krazr);
//c  pause '8'
      izw=izl;
      kcw=kcrez;
      kdw=kdrez;
      for(lp=1;lp<=kcw;lp++)
        icw[lp]=icrez[lp];
      for(lp=1;lp<=kdw;lp++)
        idw[lp]=idrez[lp];
//16102      continue
  }
m16101:;
//      fprintf(fp6,"\n m1601\n");
//      scanf("%d",&igur);
      istep=k-(s-1);
      izw1=1;
      kcw1=1;
      kdw1=1;
      icw1[1]=1;
      idw1[1]=0;
      izr=izx[(i-1)*m+p];
      kcr=kcx[(i-1)*m+p];
      kdr=kdx[(i-1)*m+p];
      for(lp=1;lp<=kcr;lp++)
        icr[lp]=icx[((i-1)*m+p-1)*k50+lp];
      for(lp=1;lp<=kdr;lp++)
        idr[lp]=idx[((i-1)*m+p-1)*k50+lp];
      if(istep==0) goto m18301;

```

```

// do 18302 lpr=1,istep
for(lpr=1;lpr<=istep;lpr++)
{

//c pause '9'
ymnzn(izw1,izr,&izrez,kcw1,kdw1,icw1,idw1,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
//c pause '10'
izw1=izrez;
kcw1=kcrez;
kdwl=kdrez;
for(lp=1;lp<=kcrez;lp++)
icw1[lp]=icrez[lp];
for(lp=1;lp<=kdrez;lp++)
idw1[lp]=idrez[lp];

//18302 continue
}

m18301:;
// fprintf(fp6,"\nm18301\n");
// scanf("%d",&igur);
//c pause '11'
// fprintf(fp6,"\n per ymnzn...\n");
ymnzn(izw,izw1,&izrez,kcw,kdw,icw,idw,
kcw1,kdw1,icw1,idw1,
&kcrez,&kdrez,icrez,idrez,krazr);
// fprintf(fp6,"\n 12\n");
//c pause '12'
kmat=k*m+1;
// fprintf(fp6,"\nmnk kmat=%d\n",kmat);
// scanf("%d",&igur);
// fprintf(fp6,"\nizc[]=%d kcc[]=%d kdc[]=%d\n",
// izc[(ilc-1)*kmat+jlc],kcc[(ilc-1)*kmat+jlc],
// kdc[(ilc-1)*kmat+jlc]);
// fprintf(fp6,"\n ilc=%d jlc=%d\n",ilc,jlc);
izr=izc[(ilc-1)*kmat+jlc];
kcr=kcc[(ilc-1)*kmat+jlc];
kdr=kdc[(ilc-1)*kmat+jlc];
// fprintf(fp6,"\n izr=%d kcr=%d kdr=%d\n",izr,kcr,kdr);
for(lp=1;lp<=kcr;lp++)
icr[lp]=icc[((ilc-1)*kmat+jlc-1)*k50+lp];
for(lp=1;lp<=kdr;lp++)
idr[lp]=idc[((ilc-1)*kmat+jlc-1)*k50+lp];
// fprintf(fp6,"\n 13\n");
// scanf("%d",&igur);
//c pause '13'
// fprintf(fp6,"\nper sl.....\n");
slzn(izrez,izr,&izl,kcrez,kdrez,icrez,idrez,
kcr,kdr,icr,idr,

&kcl,&kdl,icl,idl,krazr);
// fprintf(fp6,"\n 14\n");
// scanf("%d",&igur);
//c pause '14'
izc[(ilc-1)*kmat+jlc]=izl;
kcc[(ilc-1)*kmat+jlc]=kcl;
kdc[(ilc-1)*kmat+jlc]=kdl;
for(lp=1;lp<=kcl;lp++)
icc[((ilc-1)*kmat+jlc-1)*k50+lp]=icl[lp];
for(lp=1;lp<=kdl;lp++)
idc[((ilc-1)*kmat+jlc-1)*k50+lp]=idl[lp];

```

```

//3  continue
}
//  fprintf(fp6, "\n 3\n");
//  scanf("%d", &igur);
istr=(p-1)*k+s;
isto=(j-1)*k+1;
//211  continue
}
//  fprintf(fp6, "\n 211\n");
//212  continue
}
//  fprintf(fp6, "\n212\n");
//  scanf("%d", &igur);
ilc=(p-1)*k+s;
jlc=(m-1)*k+k+1;
izc[(ilc-1)*kmat+jlc]=1;
kcc[(ilc-1)*kmat+jlc]=1;
kdc[(ilc-1)*kmat+jlc]=1;
icc[((ilc-1)*kmat+jlc-1)*k50+1]=0;
idc[((ilc-1)*kmat+jlc-1)*k50+1]=0;
//c  c((p-1)*k+s, (m-1)*k+k+1)=0
//  do 703 i=1,n
for(i=1;i<=n;i++)
{
//  fprintf(fp6, "\n 703 i=%d\n", i);
//  scanf("%d", &igur);
istep=k-(s-1);
izw=1;
kcw=1;
kdw=1;
icw[1]=1;
idw[1]=0;
izr=izx[(i-1)*m+p];
kcr=kcx[(i-1)*m+p];
kdr=kdx[(i-1)*m+p];
for(lp=1;lp<=kcr;lp++)
icr[lp]=icx[((i-1)*m+p-1)*k50+lp];
for(lp=1;lp<=kdr;lp++)
idr[lp]=idx[((i-1)*m+p-1)*k50+lp];
if(istep==0) goto m18403;
//  do 18404 lpr=1, istep
for(lpr=1;lpr<=istep;lpr++)
{
//c  pause '15'
ymnzn(izw,izr, &izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
//c  pause '16'
izw=izrez;
kcw=kcrez;
kdw=kdrez;
for(lp=1;lp<=kcrez;lp++)

icw[lp]=icrez[lp];
for(lp=1;lp<=kdrez;lp++)
idw[lp]=idrez[lp];
//18404  continue
}

m18403:;
//  fprintf(fp6, "\n m18403\n");
//  scanf("%d", &igur);
izr=izc[(ilc-1)*kmat+jlc];

```

```

kcr=kcc[(i1c-1)*kmat+j1c];
kdr=kdc[(i1c-1)*kmat+j1c];
for(lp=1;lp<=kcr;lp++)
icr[lp]=icc[((i1c-1)*kmat+j1c-1)*k50+lp];
for(lp=1;lp<=kdr;lp++)
idr[lp]=idc[((i1c-1)*kmat+j1c-1)*k50+lp];
// printf("\npered slzn\n");
// scanf("%d",&igur);
// slzn(izw,izr,&izrez,kcw,kdw,icw,idw,
kcr,kdr,icr,idr,
&kcrez,&kdrez,icrez,idrez,krazr);
// printf("\nposle slzn\n");
// scanf("%d",&igur);
izc[(i1c-1)*kmat+j1c]=izrez;
kcc[(i1c-1)*kmat+j1c]=kcrez;
kdc[(i1c-1)*kmat+j1c]=kdrez;
for(lp=1;lp<=kcrez;lp++)
icc[((i1c-1)*kmat+j1c-1)*k50+lp]=icrez[lp];
for(lp=1;lp<=kdrez;lp++)
idc[((i1c-1)*kmat+j1c-1)*k50+lp]=idrez[lp];
//c c((p-1)*k+s,(m-1)*k+k+1)=c((p-1)*k+s,(m-1)*k+k+1)+
//c *x(i,p)**(k-(s-1))
// printf("\npered 703\n");
// scanf("%d",&igur);
//703 continue
}
// fprintf(fp6,"\nposle 703\n");
// scanf("%d",&igur);
istr=(p-1)*k+s;
isto=(m-1)*k+k+1;
//c write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)
//282 continue
}
//2 continue
}
// fprintf(fp6,"\n posle 2\n");
// scanf("%d",&igur);
inf=(m-1)*k+k+1;
izf[inf]=1;
kcf[inf]=1;
kdf[inf]=1;
icf[(inf-1)*k50+1]=0;
idf[(inf-1)*k50+1]=0;
//c f((m-1)*k+k+1)=0
// do 12002 i=1,n
for(i=1;i<=n;i++)
{
izr=izf[inf];
kcr=kcf[inf];
kdr=kdf[inf];
for(lp=1;lp<=kcr;lp++)
icr[lp]=icf[(inf-1)*k50+lp];
for(lp=1;lp<=kdr;lp++)
idr[lp]=idf[(inf-1)*k50+lp];
iz1=izy[i];
kcl=kcy[i];
kdl=kdy[i];
for(lp=1;lp<=kcl;lp++)
ic1[lp]=icy[(i-1)*k50+lp];
for(lp=1;lp<=kdl;lp++)
id1[lp]=idy[(i-1)*k50+lp];

slzn(izr,iz1,&izrez,kcr,kdr,icr,idr,

```

```

kcl,kdl,icl,idl,
&kcrez,&kdrez,icrez,idrez,krazr);
  izf[inf]=izrez;
  kcf[inf]=kcrez;
  kdf[inf]=kdrez;
  for(lp=1;lp<=kcrez;lp++)
  icf[(inf-1)*k50+lp]=icrez[lp];
  for(lp=1;lp<=kdrez;lp++)
  idf[(inf-1)*k50+lp]=idrez[lp];

//c  f((m-1)*k+k+1)= f((m-1)*k+k+1)+y(i)
//12002  continue
  }
//  fprintf(fp6,"\npered 4212\n");
//  do 4212 j=1,m
//    for(j=1;j<=m;j++)
//      {
//        do 4211 l=1,k
//          for(l=1;l<=k;l++)
//            {
//              fprintf(fp6,"\n 4211 j=%d l=%d\n",j,l);
//              icc1=(m-1)*k+k+1;
//              jc1=(j-1)*k+1;
//              izc[(icc1-1)*kmat+jc1]=1;
//              kcc[(icc1-1)*kmat+jc1]=1;
//              kdc[(icc1-1)*kmat+jc1]=1;
//              icc[(icc1-1)*kmat+jc1-1]*k50+1]=0;
//              idc[(icc1-1)*kmat+jc1-1]*k50+1]=0;
//c      c((m-1)*k+k+1,(j-1)*k+1)=0
//      do 371 i=1,n
//        for(i=1;i<=n;i++)
//          {
//            fprintf(fp6,"\n 371 i=%d\n",i);
//            istep=k-(l-1);
//            izw=1;
//            kcw=1;
//            kdw=1;
//            icw[1]=1;
//            idw[1]=0;
//            izr=izx[(i-1)*m+j];
//            kcr=kcx[(i-1)*m+j];
//            kdr=kdx[(i-1)*m+j];
//            fprintf(fp6,"\n371 izr=%d kcr=%d kdr=%d\n",
//            izr,kcr,kdr);
//            for(lp=1;lp<=kcr;lp++)
//              icr[lp]=icx[((i-1)*m+j-1)*k50+lp];
//            for(lp=1;lp<=kdr;lp++)
//              idr[lp]=idx[((i-1)*m+j-1)*k50+lp];
//            fprintf(fp6,"\n 371 istep=%d\n",istep);
//            if(istep==0) goto m23673;
//            do 23674 lpr=1,istep
//              for(lpr=1;lpr<=istep;lpr++)
//                {
//                  fprintf(fp6,"\n 371 lpr=%d izr=%d kcr=%d kdr=%d\n",
//                  lpr,izr,kcr,kdr);
//c          pause '21'
//          ymnzn(izw,izr,&izrez,kcw,kdw,icw,idw,
//          kcr,kdr,icr,idr,
//          &kcrez,&kdrez,icrez,idrez,krazr);
//          fprintf(fp6,"\n 371 posle ymnzn\n");
//          if(l==2) return;
//          izw=izrez;
//          kcw=kcrez;

```

```

        kdw=kdrez;
//fprintf(fp6,"\n 371 izw=%d kcw=%d kdw=%d\n",izw,kcw,kdw);
//    if(l==2) return;
        for(lp=1;lp<=kcrez;lp++)
            icw[lp]=icrez[lp];
        for(lp=1;lp<=kdrez;lp++)

            idw[lp]=idrez[lp];

//23674    continue
    }
m23673:;
//    fprintf(fp6,"\nm23673 \n");
//    if(l==2) return;
        izr=izc[(iccl-1)*kmat+jc1];
        kcr=kcc[(iccl-1)*kmat+jc1];
        kdr=kdc[(iccl-1)*kmat+jc1];
        for(lp=1;lp<=kcr;lp++)
            icr[lp]=icc[((iccl-1)*kmat+jc1-1)*k50+lp];
        for(lp=1;lp<=kdr;lp++)
            idr[lp]=idc[((iccl-1)*kmat+jc1-1)*k50+lp];
        slzn(izw,izr,&izrez,kcw,kdw,icw,idw,
            kcr,kdr,icr,idr,
            &kcrez,&kdrez,icrez,idrez,krazr);

        izc[(iccl-1)*kmat+jc1]=izrez;
        kcc[(iccl-1)*kmat+jc1]=kcrez;
        kdc[(iccl-1)*kmat+jc1]=kdrez;
        for(lp=1;lp<=kcrez;lp++)
            icc[((iccl-1)*kmat+jc1-1)*k50+lp]=icrez[lp];
        for(lp=1;lp<=kdrez;lp++)
            idc[((iccl-1)*kmat+jc1-1)*k50+lp]=idrez[lp];
//c    c((m-1)*k+k+1,(j-1)*k+1)= c((m-1)*k+k+1,(j-1)*k+1)+
//c        *x(i,j)**(k-(l-1))
//371    continue
    }
//    fprintf(fp6,"\n 371\n");
//    if(l==2) return;
        istr=(m-1)*k+k+1;
        isto=(j-1)*k+1;
//4211    continue
    }
//4212    continue
    }
//    fprintf(fp6,"\n4212 \n");
//    if(k==2) return;
        iccl=(m-1)*k+k+1;
        izc[(iccl-1)*kmat+iccl]=izn;
        kcc[(iccl-1)*kmat+iccl]=kcn;
        kdc[(iccl-1)*kmat+iccl]=kdn;
        for(lp=1;lp<=kcn;lp++)
            icc[((iccl-1)*kmat+iccl-1)*k50+lp]=icn[lp];
        for(lp=1;lp<=kdn;lp++)
            idc[((iccl-1)*kmat+iccl-1)*k50+lp]=idn[lp];

//c    c((m-1)*k+k+1,(m-1)*k+k+1)=n

        istr=(m-1)*k+k+1;
        isto=(m-1)*k+k+1;
//c    write(6,3311) p,s,j,l,istr,isto,c(istr,isto),f(istr)

```

```

    kraz=m*k+1;
//   fprintf(fp6,"\n zap c\n");
//   if(k==2) return;
//c   write(5,*) '   матрица c'
//   open(9,file='promprow',status='Unknown')
    fp9=fopen("promprc","w");
    fprintf(fp9,"%d\n",kraz);
//   do 56 i=1,kraz
    for(i=1;i<=kraz;i++)
    {
//   do 1234 j=1,kraz
    for(j=1;j<=kraz;j++)
    {
        izr=izc[(i-1)*kmat+j];
        kcel=kcc[(i-1)*kmat+j];
        kdel=kdc[(i-1)*kmat+j];

        fprintf(fp9,"%d %d %d \n",izr,kcel,kdel);
        for(jp=1;jp<=kcel;jp++)
            fprintf(fp9,"%d ",icc[((i-1)*kmat+j-1)*k50+jp]);
        fprintf(fp9,"\n");

        for(jp=1;jp<=kdel;jp++)
            fprintf(fp9,"%d ",idc[((i-1)*kmat+j-1)*k50+jp]);
        fprintf(fp9,"\n");
//1234        continue
    }
//56    continue
    }
//   do 56701 i=1,kraz
    for(i=1;i<=kraz;i++)
    {
        izr=izf[i];
        kcel=kcf[i];
        kdel=kdf[i];

        fprintf(fp9,"%d %d %d\n", izr,kcel,kdel);
        for(ip=1;ip<=kcel;ip++)
            fprintf(fp9,"%d ",icf[(i-1)*k50+ip]);
        fprintf(fp9,"\n");
        for(ip=1;ip<=kdel;ip++)

            fprintf(fp9,"%d ",idf[(i-1)*k50+ip]);
        fprintf(fp9,"\n");
    }
//   close(9,status='keep')
    fclose(fp9);
    kg=k*m+1;
//c   call gay(kg,c,f,ak,igau)
//c   pause 'pered gaydl'
//   fprintf(fp6,"\npered gaydl\n");
//   if(k==2) return;
//   scanf("%d",&igur);
    gaydl(argc,argv,kg,izc,kcc,kdc,icc,idc,izf,kcf,kdf,icf,idf,
izak,kcak,kdak,icak,idak,&igau,krazr,k50,mnper);
    *pigau=igau;
//   fprintf(fp6,"\n posle gaydl\n");
//   if(k==2) return;
//   scanf("%d",&igur);
    if(igau==0) return;
//c   pause 'posle gaydl'
    kg=k*m+1;
//   fprintf(fp6,"\npered prowg\n");

```

```

//  if(k==2) return;
//  prorg(kg,izc,kcc,kdc,icc,idc,
//  izf,kcf,kdf,icf,idf,izak,kcak,kdak,icak,idak,&igau,krazr,k50);
//  fprintf(fp6,"\n posle prorg\n");
//  if(k==2) return;
//  return;
//  }

//Набор gaydl.c(модификация для распараллеливания)

#include<stdio.h>
#include<math.h>
#include<alloc.h>

/*
с-метод Гаусса решения систем с выбором главного элемента,длинная арифметика
с-прямой ход метода Гаусса распараллелен
с-ac(i,j,l)-целая часть элемента i,j матрицы системы
с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
с-nc(I,j)-количество разрядов целой части элемента I,j
с-nd(I,j)-количество разрядов дробной части элемента I,j
с-znak(I,j)-знак элемента I,j
с-bc(I,l)-целая часть правой части i-ого уравнения
с-bd(I,l)-дробная часть правой части i-го уравнения
с-nbc(i)-длина целой части правой части i-го уравнения
с-nbd(i)-длина дробной части правой части i-го уравнения
с-znp(i)-знак правой части i-ого уравнения
с-xc(I,l)-целая часть i-го неизвестного
с-xd(I,l)-дробная часть i-го неизвестного
с-nxc(i)-длина целой части i-ого неизвестного
с-nxd(i)-длина дробной части i-ого неизвестного
с-znx(i)-знак i-ой неизвестной */
void gaydl(int argc, char *argv[ ],int n,int *znak,
int *nc,int *nd,
int *ac,int *ad,
int *znp,int *nbc,int *nbd,
int *bc,int *bd,
int *znx,int *nxc,int *nxd,
int *xc,int *xd,
int *pigau,int krazr,int k50,int *mnper)
{
//  extern FILE *fp6;

extern void obrdl(int n,int *znak,
int *nc,int *nd,
int *ac,int *ad,
int *znp,int *nbc,int *nbd,
int *bc,int *bd,
int *znx,int *nxc,int *nxd,
int *xc,int *xd,
int *pigau,int *mnper,int anc[400],
int andr[400],int krazr,int k50);

extern void nimax(int i,int j,int n,
int *nc,int *nd, int *znak,
int *ac,int *ad,
int *nbc,int *nbd, int *znp,
int *bc,int *bd,
int *mnper, int krazr,int k50);

```

```

extern void delzn(int izr,int iznt,int *pizrez,int ncr,
int ndr,int *rc,int *rd,
int nct,int ndt,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void ymnzn(int izp,int izr,int *pizrez,int ncp,int ndp,
int *pc,int *pd,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void wiczn(int izrl,int izr,int *pizrez,int ncr1,int ndr1,
int *rc1,int *rd1,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);
int myid, numprocs;
int nt,ipoi,jpoi,jper,imax,tag,ipos,
jmax,ipa,ikonp,nper,nproc;
int peres[51];

MPI_Status status;
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
MPI_Comm_rank(MPI_COMM_WORLD, &myid);

int anc[400],andr[400],rezc[400],rezd[400],
*pizrez,*pkcrez,*pkdrez,igau,
pc[400],pd[400],rc1[400],rd1[400];
int rc[400],rd[400],tc[400],td[400];
int i,nis,j,nct,ndt,iznt,ld,nolc,nold,iznol,l,ncl,ndl,
izn1,js,izt,nlc,nld,ncr,ndr,izr,kcrez,kdrez,izrez,k,ncp,ndp,
izp,ncr1,ndr1,izrl,zn,ncl,ndl,izn1;
// С двух следующих строчек убрать комментарии, если не будет получаться,

//две таких строчки закомментировать после start parallel внутри
// if(myid==0)
// {
peres[0]=0;

for(i=1;i<=n;i++)
mnp[per[i]]=i;
if(krazr<=k50) goto m8855;
if(myid==0)
printf(" количество разрядов в дробной части>%d,нельзя!\n",k50);
*pigau=0;

return;
m8855:;
nis=n;
i=1;
m81: j=i;
nimax(i,j,n,nc,nd,znak,ac,ad,nbc,nbd,znp,bc,bd,
mnp[per[i],krazr,k50);
// fprintf(fp6,"\nпосле nimax\n");
//c t=a(i,i)
nct=nc[(i-1l)*n+i];
ndt=nd[(i-1l)*n+i];
iznt=znak[(i-1l)*n+i];
for(ld=1;ld<=nct;ld++)

```

```

tc[ld]=ac[ ((i-1l)*n+i-1)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[ ((i-1l)*n+i-1)*k50+ld];
for(ld=1;ld<=399;ld++)
{
anc[ld]=0;
andr[ld]=0;
}
nolc=399;
nold=399;
iznol=1;
for(ld=1;ld<=nct;ld++)
if(tc[ld]!=anc[ld]) goto m2;
for(ld=1;ld<=ndt;ld++)

if(td[ld]!=andr[ld]) goto m2;
l=i+1;
m4:;
ncl=nc[ (l-1l)*n+j];
ndl=nd[ (l-1l)*n+j];
iznl=znak[ (l-1l)*n+j];
for(ld=1;ld<=ncl;ld++)
if(ac[ ((l-1l)*n+j-1)*k50+ld]!=anc[ld]) goto m3;
for(ld=1;ld<=ndl;ld++)
if(ad[ ((l-1l)*n+j-1)*k50+ld]!=andr[ld]) goto m3;
l=l+1;
if(l<=n) goto m4;
if(myid==0) printf(" матрица вырожденная \n");

igau=0;
*pigau=igau;
return;
m3:;
for(jjs=1;jjs<=n;jjs++)
{
//c tz=a(l,jjs)
//c a(l,jjs)=a(i,jjs)
//c a(i,jjs)=tz
nct=nc[ (l-1l)*n+jjs];
ndt=nd[ (l-1l)*n+jjs];
izt=znak[ (l-1l)*n+jjs];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[ ((l-1l)*n+jjs-1)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[ ((l-1l)*n+jjs-1)*k50+ld];
nc[ (l-1l)*n+jjs]=nc[ (i-1l)*n+jjs];
nd[ (l-1l)*n+jjs]=nd[ (i-1l)*n+jjs];

znak[ (l-1l)*n+jjs]=znak[ (i-1l)*n+jjs];
n1c=nc[ (i-1l)*n+jjs];
n1d=nd[ (i-1l)*n+jjs];
for(ld=1;ld<=n1c;ld++)
ac[ ((l-1l)*n+jjs-1)*k50+ld]=ac[ ((i-1l)*n+jjs-1)*k50+ld];
for(ld=1;ld<=n1d;ld++)
ad[ ((l-1l)*n+jjs-1)*k50+ld]=ad[ ((i-1l)*n+jjs-1)*k50+ld];
nc[ (i-1l)*n+jjs]=nct;
nd[ (i-1l)*n+jjs]=ndt;
znak[ (i-1l)*n+jjs]=izt;
for(ld=1;ld<=nct;ld++)
ac[ ((i-1l)*n+jjs-1)*k50+ld]=tc[ld];

```

```

for (ld=1;ld<=ndt;ld++)
ad[ ((i-1l)*n+js-1)*k50+ld]=td[ld];
}
nct=NBC[i];
ndt=NDI[i];
izt=ZNP[i];
for (ld=1;ld<=nct;ld++)
tc[ld]=bc[ (i-1l)*k50+ld];
for (ld=1;ld<=ndt;ld++)
td[ld]=bd[ (i-1l)*k50+ld];
NBC[i]=NBC[l];
NDI[i]=NDI[l];
ZNP[i]=ZNP[l];
n1c=NBC[l];
n1d=NDI[l];
for (ld=1;ld<=n1c;ld++)

bc[ (i-1l)*k50+ld]=bc[ (l-1l)*k50+ld];
for (ld=1;ld<=n1d;ld++)
bd[ (i-1l)*k50+ld]=bd[ (l-1l)*k50+ld];
NBC[l]=nct;
NDI[l]=ndt;
ZNP[l]=izt;
for (ld=1;ld<=nct;ld++)
bc[ (l-1l)*k50+ld]=tc[ld];
for (ld=1;ld<=ndt;ld++)
bd[ (l-1l)*k50+ld]=td[ld];
nct=nc[ (i-1l)*n+i];
ndt=nd[ (i-1l)*n+i];
iznt=znak[ (i-1l)*n+i];
for (ld=1;ld<=nct;ld++)
tc[ld]=ac[ ((i-1l)*n+i-1)*k50+ld];
for (ld=1;ld<=ndt;ld++)
td[ld]=ad[ ((i-1l)*n+i-1)*k50+ld];
//c t=a(i,i)
m2;;
m2222;;
//c a(i,j)=a(i,j)/t
ncr=nc[ (i-1l)*n+j];
ndr=nd[ (i-1l)*n+j];
izr=znak[ (i-1l)*n+j];
for (ld=1;ld<=ncr;ld++)
rc[ld]=ac[ ((i-1l)*n+j-1)*k50+ld];

for (ld=1;ld<=ndr;ld++)
rd[ld]=ad[ ((i-1l)*n+j-1)*k50+ld];
// fprintf(fp6, "\n прямой ход, деление левой части a(i,j)/t \n");

// fprintf(fp6, "делимое a(i,j)\n");
// fprintf(fp6, "i=%d j=%d\n", i, j);

// fprintf(fp6, "знак=%d\n", izr);
// fprintf(fp6, "длина целой части=%d длина дробной части=%d\n", ncr, ndr);
/* fprintf(fp6, "знак=%d\n", izr);
fprintf(fp6, "длина целой части=%d длина дробной части=%d\n", ncr, ndr);
fprintf(fp6, "\n целая часть\n");
for (ld=1;ld<=ncr;ld++)
fprintf(fp6, "%ld", rc[ld]);
fprintf(fp6, "\n дробная часть\n");
for (ld=1;ld<=ndr;ld++)
fprintf(fp6, "%ld", rd[ld]);

```

```

    fprintf(fp6, "\n делитель t \n");
    fprintf(fp6, "знак=%d\n", iznt);
    fprintf(fp6, "длина целой части=%d длина дробной части=%d\n", nct, ndt);
    fprintf(fp6, "\n целая часть\n");
    for(ld=1; ld<=nct; ld++)
        fprintf(fp6, "%ld", tc[ld]);
    fprintf(fp6, "\n дробная часть\n");
    for(ld=1; ld<=ndt; ld++)
        fprintf(fp6, "%ld", td[ld]); */
    delzn(izr, iznt, &izrez, ncr, ndr, rc, rd, nct, ndt, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
/*    fprintf(fp6, "\n частное a(i, j)/t \n");
    fprintf(fp6, "знак=%d\n", izrez);
    fprintf(fp6, "длина целой части=%d длина дробной части=%d\n",
        kcrez, kdrez);
    fprintf(fp6, "\n целая часть\n");

    for(ld=1; ld<=kcrez; ld++)
        fprintf(fp6, "%ld", rezc[ld]);
    fprintf(fp6, "\n дробная часть\n");
    for(ld=1; ld<=kdrez; ld++)
        fprintf(fp6, "%ld", rezd[ld]);

    fprintf(fp6, "\nprjm после delzn a/t\n"); */
    nc[(i-1l)*n+j]=kcrez;
    nd[(i-1l)*n+j]=kdrez;
    znak[(i-1l)*n+j]=izrez;
    for(ld=1; ld<=kcrez; ld++)
        ac[((i-1l)*n+j-1)*k50+ld]=rezc[ld];
    for(ld=1; ld<=kdrez; ld++)
        ad[((i-1l)*n+j-1)*k50+ld]=rezd[ld];
    j=j+1;
    if(j<=n) goto m2222;
//c    b(i)=b(i)/t
        ncr=nc[i];
        ndr=nd[i];
        izr=zn[i];
        for(ld=1; ld<=ncr; ld++)
            rc[ld]=bc[(i-1l)*k50+ld];
        for(ld=1; ld<=ndr; ld++)
            rd[ld]=bd[(i-1l)*k50+ld];
        delzn(izr, iznt, &izrez, ncr, ndr, rc, rd, nct, ndt, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
//    fprintf(fp6, "\nprjm после delzn b/t\n");
        nbc[i]=kcrez;
        nbd[i]=kdrez;

        zn[i]=izrez;
        for(ld=1; ld<=kcrez; ld++)
            bc[(i-1l)*k50+ld]=rezc[ld];
        for(ld=1; ld<=kdrez; ld++)
            bd[(i-1l)*k50+ld]=rezd[ld];
//отсюда вычитание i-ого из k-ого
//    fprintf(fp5, " b после %30.14f\n", b[i]);
// start parallel
//MPI_Status status;

// MPI_Init(&argc, &argv);
//MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
//MPI_Comm_rank(MPI_COMM_WORLD, &myid);
        peres[0]=0;

```

```
// Не будет получаться-закомментировать две следующие строчки,
// а в самом начале комментария с этих строчек убрать
```

```

if(myid==0)
{
MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&i, 1, MPI_INT, 0, MPI_COMM_WORLD);
MPI_Bcast(&numprocs, 1, MPI_INT, 0, MPI_COMM_WORLD);
peres[0]=0;
for(ipa=1;ipa<numprocs;ipa++)
{
tag=0;
tag=tag+1;
for(j=1;j<=n;j++)
peres[j]= nc[(i-1l)*n+j];
nper=n+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);
tag=tag+1;
for(j=1;j<=n;j++)
peres[j]= nd[(i-1l)*n+j];
nper=n+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);
tag=tag+1;
for(j=1;j<=n;j++)
peres[j]= znak[(i-1l)*n+j];
nper=n+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);

for(j=1;j<=n;j++)
{
nct=nc[(i-1l)*n+j];
ndt=nd[(i-1l)*n+j];
iznt=znak[(i-1l)*n+j];

for(ld=1;ld<=nct;ld++)
peres[ld]=ac[((i-1l)*n+j-1)*k50+ld];
tag=tag+1;
nper=nct+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);

for(ld=1;ld<=ndt;ld++)

peres[ld]=ad[((i-1l)*n+j-1)*k50+ld];
tag=tag+1;
nper=ndt+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);
}
}

```

```

// for(j=1;j<=n;j++)
// printf("process=0 processy=%3d soob=%3d i=%3d \n",
// ipa,tag,i);
// printf("\n");
for(ld=1;ld<=n;ld++)
peres[ld]=nbc[ld];

```

```

    tag=tag+1;
    nper=n+1;

    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);
    for(ld=1;ld<=n;ld++)
    peres[ld]=nbd[ld];
    tag=tag+1;
    nper=n+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);
    for(ld=1;ld<=n;ld++)
    peres[ld]=znp[ld];
    tag=tag+1;
    nper=n+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);
    for(j=1;j<=n;j++)
    {
    ncr=ncb[j];
    ndr=nbd[j];
    izr=znp[j];
    for(ld=1;ld<=ncr;ld++)
    peres[ld]=bc[(j-1)*k50+ld];
    tag=tag+1;
    nper=ncr+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);

    for(ld=1;ld<=ndr;ld++)
    peres[ld]=bd[(j-1)*k50+ld];
    tag=tag+1;
    nper=ndr+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);
    }
    for(k=i+ipa;k<=n;k+=numprocs-1)
    {
//    for(j=1;j<=n;j++)

//    peres[j]=a[k][j];
    for(j=1;j<=n;j++)
    peres[j]=nc[(k-1)*n+j];
    nper=n+1;
    tag=tag+1;

    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);

    tag=tag+1;
    for(j=1;j<=n;j++)
    peres[j]=nd[(k-1)*n+j];
    nper=n+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);
    tag=tag+1;
    for(j=1;j<=n;j++)
    peres[j]=znak[(k-1)*n+j];
    nper=n+1;
    MPI_Send(peres,nper,MPI_INT,ipa,tag,MPI_COMM_WORLD);

    for(j=1;j<=n;j++)
    {
    nct=nc[(k-1)*n+j];
    ndt=nd[(k-1)*n+j];
    iznt=znak[(k-1)*n+j];

```

```

    for(ld=1;ld<=nct;ld++)
peres[ld]=ac[ ((k-11) *n+j-1) *k50+ld];
tag=tag+1;
nper=nct+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);

for(ld=1;ld<=ndt;ld++)

    peres[ld]=ad[ ((k-11) *n+j-1) *k50+ld];
tag=tag+1;
nper=ndt+1;
MPI_Send(peres,nper, MPI_INT,ipa,tag,MPI_COMM_WORLD);
}

/* end po k */
/* end po ipa */
// priem
for(ipa=1;ipa<numprocs;ipa++)
{
ipos=0;
for(k=i+ipa;k<=n;k+= numprocs-1)
{
nper=n+1;
ipos=ipos+1;
MPI_Recv(peres,nper, MPI_INT,ipa,ipos,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
nc[(k-11) *n+j]= peres[j];

ipos=ipos+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,ipa,ipos,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
nd[(k-11) *n+j]= peres[j];

ipos=ipos+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,ipa,ipos,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
znak[(k-11) *n+j]= peres[j];

for(j=1;j<=n;j++)
{
nct=nc[(k-11) *n+j];
ndt=nd[(k-11) *n+j];
iznt=znak[(k-11) *n+j];

ipos=ipos+1;
nper=nct+1;
MPI_Recv(peres,nper, MPI_INT,ipa,ipos,MPI_COMM_WORLD,&status);

for(ld=1;ld<=nct;ld++)
ac[ ((k-11) *n+j-1) *k50+ld]= peres[ld];

ipos=ipos+1;
nper=ndt+1;

```

```

MPI_Recv(peres,nper, MPI_INT,ipa,ipos,MPI_COMM_WORLD,&status);
for(ld=1;ld<=ndt;ld++)

ad[(k-1l)*n+j-1]*k50+ld]= peres[ld];

}

    ipos=ipos+1;
    nper=n+1;
MPI_Recv(peres,nper,MPI_INT,ipa,ipos, MPI_COMM_WORLD,&status);
// for(ld=1;ld<=n;ld++)
    ld=k;
    nbc[ld]= peres[ld];

    ipos=ipos+1;
    nper=n+1;
MPI_Recv(peres,nper,MPI_INT,ipa,ipos, MPI_COMM_WORLD,&status);
// for(ld=1;ld<=n;ld++)
    ld=k;
    nbd[ld]= peres[ld];

    ipos=ipos+1;
    nper=n+1;
MPI_Recv(peres,nper,MPI_INT,ipa,ipos, MPI_COMM_WORLD,&status);
// for(ld=1;ld<=n;ld++)
    ld=k;
    znp[ld]= peres[ld];

// for(j=1;j<=n;j++)
// {
    j=k;
    ncr=nbc[j];
    ndr=nbd[j];
    izr=znp[j];

    ipos=ipos+1;
    nper=ncr+1;
MPI_Recv(peres,nper,MPI_INT,ipa,ipos, MPI_COMM_WORLD,&status);
for(ld=1;ld<=ncr;ld++)
bc[(j-1l)*k50+ld]= peres[ld];

    ipos=ipos+1;
    nper=ndr+1;
MPI_Recv(peres,nper,MPI_INT,ipa,ipos, MPI_COMM_WORLD,&status);
for(ld=1;ld<=ndr;ld++)
bd[(j-1l)*k50+ld]= peres[ld];

// }

/* end po k */
/* end po ipa */
/* end myid=0 */
    else
    { /* myid!=0 */

ipa=myid;

```

```

tag=1;
ipos=0;
nproc=myid;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
nc[(i-1l)*n+j]=peres[j];

tag=tag+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);

for(j=1;j<=n;j++)
nd[(i-1l)*n+j]= peres[j];

tag=tag+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
znak[(i-1l)*n+j]= peres[j];

for(j=1;j<=n;j++)
{
nct=nc[(i-1l)*n+j];
ndt=nd[(i-1l)*n+j];

tag=tag+1;
nper=nct+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);

for(ld=1;ld<=nct;ld++)
ac[((i-1l)*n+j-1)*k50+ld]= peres[ld];

tag=tag+1;
nper=ndt+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for(ld=1;ld<=ndt;ld++)

ad[((i-1l)*n+j-1)*k50+ld]= peres[ld];
}

// for(j=1;j<=n;j++)
// printf("priem w processe=%3d TAG =%3d i=%3d \n ",
// ipa,tag,i);
// printf("\n");

tag=tag+1;
nper=n+1;
MPI_Recv(peres,nper,MPI_INT,0,tag, MPI_COMM_WORLD,&status);

```

```

for (ld=1;ld<=n;ld++)
nbc[ld]= peres[ld];

tag=tag+1;
nper=n+1;

MPI_Recv(peres,nper,MPI_INT,0,tag, MPI_COMM_WORLD,&status);
for (ld=1;ld<=n;ld++)
nbd[ld]= peres[ld];

tag=tag+1;
nper=n+1;
MPI_Recv(peres,nper,MPI_INT,0,tag, MPI_COMM_WORLD,&status);

for (ld=1;ld<=n;ld++)
znp[ld]= peres[ld];

for (j=1;j<=n;j++)
{
ncr=nbc[j];
ndr=nbd[j];
izr=znp[j];

tag=tag+1;
nper=ncr+1;
MPI_Recv(peres,nper,MPI_INT,0,tag, MPI_COMM_WORLD,&status);
for (ld=1;ld<=ncr;ld++)
bc[(j-1)*k50+ld]= peres[ld];

tag=tag+1;
nper=ndr+1;
MPI_Recv(peres,nper,MPI_INT,0,tag, MPI_COMM_WORLD,&status);
for (ld=1;ld<=ndr;ld++)
bd[(j-1)*k50+ld]= peres[ld];
}

// for (j=1;j<=n;j++)

// printf("priem w processe=%3d TAG =%3d i=%3d \n ",
// ipa,tag,i);

for (k=i+ipa;k<=n;k+= numprocs-1)
{
tag=tag+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for (j=1;j<=n;j++)
nc[(k-1)*n+j]= peres[j];

tag=tag+1;

nper=n+1;

```

```

MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);

for(j=1;j<=n;j++)
nd[(k-1l)*n+j]= peres[j];

tag=tag+1;

nper=n+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for(j=1;j<=n;j++)
znak[(k-1l)*n+j]= peres[j];

for(j=1;j<=n;j++)
{
nct=nc[(k-1l)*n+j];
ndt=nd[(k-1l)*n+j];
iznt=znak[(k-1l)*n+j];

tag=tag+1;
nper=nct+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);

for(ld=1;ld<=nct;ld++)
ac(((k-1l)*n+j-1)*k50+ld)= peres[ld];

tag=tag+1;
nper=ndt+1;
MPI_Recv(peres,nper, MPI_INT,0,tag,MPI_COMM_WORLD,&status);
for(ld=1;ld<=ndt;ld++)

ad(((k-1l)*n+j-1)*k50+ld)= peres[ld];
}
m7l:;
m7l11:;
ncp=nc[(k-1l)*n+i];
ndp=nd[(k-1l)*n+i];
izp=znak[(k-1l)*n+i];
for(ld=1;ld<=ncp;ld++)
pc[ld]=ac(((k-1l)*n+i-1)*k50+ld);
for(ld=1;ld<=ndp;ld++)
pd[ld]=ad(((k-1l)*n+i-1)*k50+ld);
//c p=a(k,i)
j=i;
m7:;
//c a(k,j)=a(k,j)-p*a(i,j)
ncr=nc[(i-1l)*n+j];
ndr=nd[(i-1l)*n+j];
izr=znak[(i-1l)*n+j];
for(ld=1;ld<=ncr;ld++)
rc[ld]=ac(((i-1l)*n+j-1)*k50+ld);
for(ld=1;ld<=ndr;ld++)
rd[ld]=ad(((i-1l)*n+j-1)*k50+ld);
ymnzn(izp,izr,&izrez,ncp,ndp,pc,pd,ncr,ndr,rc,rd,
&kcrez,&kdrez,rezc,rezd,krazr);
// fprintf(fp6,"\nprjm posle ymnzn p*a(i,j)\n");
ncr=kcrez;
ndr=kdrez;
izr=izrez;
for(ld=1;ld<=ncr;ld++)

```

```

rc[ld]=rezc[ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=rezd[ld];
ncr1=nc[(k-1l)*n+j];
ndr1=nd[(k-1l)*n+j];
izr1=znak[(k-1l)*n+j];
for(ld=1;ld<=ncr1;ld++)
rc1[ld]=ac[((k-1l)*n+j-1)*k50+ld];
for(ld=1;ld<=ndr1;ld++)
rd1[ld]=ad[((k-1l)*n+j-1)*k50+ld];
wiczn(izr1,izr,&izrez,ncr1,ndr1,rc1,rd1,ncr,ndr,rc,
rd,&kcrez,&kdrez,rezc,rezd,krazr);
// fprintf(fp6,"\nprjm posle wiczn a(k,j)-p*a(i,j)\n");
nc[(k-1l)*n+j]=kcrez;
nd[(k-1l)*n+j]=kdrez;
znak[(k-1l)*n+j]=izrez;
for(ld=1;ld<=kcrez;ld++)
ac[((k-1l)*n+j-1)*k50+ld]=rezc[ld];

for(ld=1;ld<=kdrez;ld++)
ad[((k-1l)*n+j-1)*k50+ld]=rezd[ld];
j=j+1;
if(j<=n) goto m7;

for(j=1;j<=n;j++)
peres[j]= nc[(k-1l)*n+j];
nper=n+1;
ipos=ipos+1;
MPI_Send(peres,nper, MPI_INT,0,ipos,MPI_COMM_WORLD);

ipos=ipos+1;
for(j=1;j<=n;j++)
peres[j]= nd[(k-1l)*n+j];
nper=n+1;
MPI_Send(peres,nper, MPI_INT,0,ipos,MPI_COMM_WORLD);

ipos=ipos+1;
for(j=1;j<=n;j++)
peres[j]= znak[(k-1l)*n+j];
nper=n+1;
MPI_Send(peres,nper, MPI_INT,0,ipos,MPI_COMM_WORLD);

for(j=1;j<=n;j++)
{
nct=nc[(k-1l)*n+j];
ndt=nd[(k-1l)*n+j];
iznt=znak[(k-1l)*n+j];

for(ld=1;ld<=nct;ld++)
peres[ld]=ac[((k-1l)*n+j-1)*k50+ld];
ipos=ipos+1;
nper=nct+1;
MPI_Send(peres,nper, MPI_INT,0,ipos,MPI_COMM_WORLD);

for(ld=1;ld<=ndt;ld++)

peres[ld]=ad[((k-1l)*n+j-1)*k50+ld];
ipos=ipos+1;
nper=ndt+1;

```

```

MPI_Send(peres,nper, MPI_INT,0,ipos,MPI_COMM_WORLD);
}

//c b(k)=b(k)-p*b(i)
ncr=NBC[i];
ndr=nbD[i];
izr=znP[i];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(i-1l)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=bd[(i-1l)*k50+ld];
ymnzn(izp,izr,&izrez,ncp,ndp,pc,pd,ncr,ndr,rc,rd,
&kcrez,&kdrez,rezc,rezd,krazr);
// fprintf(fp6,"\nprjm posle ymnzn p*b\n");

ncr=kcrez;
ndr=kdrez;
izr=izrez;
for(ld=1;ld<=kcrez;ld++)
rc[ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
rd[ld]=rezd[ld];
ncr1=NBC[k];
ndr1=nbD[k];
izr1=znP[k];
for(ld=1;ld<=ncr1;ld++)
rc1[ld]=bc[(k-1l)*k50+ld];
for(ld=1;ld<=ndr1;ld++)
rd1[ld]=bd[(k-1l)*k50+ld];
wiczn(izr1,izr,&izrez,ncr1,ndr1,rc1,rd1,ncr,ndr,rc,rd,
&kcrez,&kdrez,rezc,rezd,krazr);
// fprintf(fp6,"\nprjm posle wiczn b(k)-p*b\n");
NBC[k]=kcrez;
nbD[k]=kdrez;
znP[k]=izrez;
for(ld=1;ld<=kcrez;ld++)
bc[(k-1l)*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
bd[(k-1l)*k50+ld]=rezd[ld];

for(ld=1;ld<=n;ld++)
peres[ld]=NBC[ld];
ipos=ipos+1;
nper=n+1;
MPI_Send(peres,nper,MPI_INT,0,ipos, MPI_COMM_WORLD);
for(ld=1;ld<=n;ld++)
peres[ld]=nbD[ld];
ipos=ipos+1;
nper=n+1;
MPI_Send(peres,nper,MPI_INT,0,ipos, MPI_COMM_WORLD);
for(ld=1;ld<=n;ld++)
peres[ld]=znP[ld];
ipos=ipos+1;
nper=n+1;
MPI_Send(peres,nper,MPI_INT,0,ipos, MPI_COMM_WORLD);
// for(j=1;j<=n;j++)
// {
j=k;
ncr=NBC[j];
ndr=nbD[j];
izr=znP[j];

```

```

for(ld=1;ld<=ncr;ld++)
peres[ld]=bc[(j-1l)*k50+ld];
  ipos=ipos+1;
nper=ncr+1;
MPI_Send(peres,nper,MPI_INT,0,ipos,MPI_COMM_WORLD);

for(ld=1;ld<=ndr;ld++)
peres[ld]=bd[(j-1l)*k50+ld];
  ipos=ipos+1;
nper=ndr+1;
MPI_Send(peres,nper,MPI_INT,0,ipos,MPI_COMM_WORLD);
// }

}/* end po k */
}/* end myid!=0 */

    i=i+1;
if(myid==0) printf(" pr i=%d\n",i);
  if(i<n) goto m8l;
if(myid==0)
obrdl(n,znak,nc,nd,ac,ad,znp,nbc,nbd,bc,bd,znx,nxc,nxd,
xc,xd,&igau,mnper,anc,andr,krazr,k50);
*pigau=igau;
// fprintf(fp6,"\n posle obr\n");
n=nis;
MPI_Finalize();
return;
}

//Набор prorg.c(модификация для распараллеливания)
// УТОЧНИТЬ РАБОТУ С ФАЙЛАМИ В UNIX,ЗДЕСЬ РАБОТА В C++ WINDOWS!
#include<stdio.h>
#include<math.h>
#include<alloc.h>

    void prorg(int n,int *znak ,
    int *nc ,int *nd ,
    int *ac ,int *ad ,
    int *znp ,int *nbc ,int *nbd ,
    int *bc ,int *bd ,
    int *znx ,int *nxc ,int *nxd ,
    int *xc ,int *xd ,
    int *pigau,int krazr,int k50)
{
extern void ymnzn(int izn1,int izn2,
int *pizrez,int ikc1,int ikd1,int *ic1,int *id1,
int ikc2,int ikd2,int *ic2,int *id2,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern void slzn(int izsum,int izn2,int *pizrez,int isc,int isd,
int *icsum,int *idsum,
int ikc2,int ikd2,
int *ic2,int *id2,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern void wiczn(int izsum,int izb,int *pizrez,int isc,int isd,
int *icsum,int *idsum,int icb,int idb,int *ic1,int *id1,
int *pkcrez,int *pkdrez,int *icrez,int *idrez,int krazr);

extern FILE *fp5,*fp9,*fp6;

//с-проверка метода Гаусса

```

```

//c-ac(i,j,l)-целая часть элемента i,j матрицы системы
//c-ad(I,j,l)-дробная часть элемента I,j матрицы системы
//c-nc(I,j)-количество разрядов целой части элемента I,j
//c-nd(I,j)-количество разрядов дробной части элемента I,j
//c-znak(I,j)-знак элемента I,j
//c-bc(I,l)-целая часть правой части i-ого уравнения
//c-bd(I,l)-дробная часть правой части i-го уравнения
//c-nbc(i)-длина целой части правой части i-го уравнения
//c-nbd(i)-длина дробной части правой части i-го уравнения
//c-znp(i)-знак правой части i-ого уравнения
//c-xc(I,l)-целая часть i-го неизвестного
//c-xd(I,l)-дробная часть i-го неизвестного
//c-nxc(i)-длина целой части i-ого неизвестного
//c-nxd(i)-длина дробной части i-ого неизвестного
//c-znx(i)-знак i-ой неизвестной
    int icel[400],idd[400],*pizrez,*pkcrez,*pkdrez,
        ic1[400],idl[400],
        ic2[400],id2[400],icrez[400],idrez[400],zn,l,
        icsum[400],idsun[400],rabl;
    int i,j,ncz,ndz,nbcz,nbdz,isc,isd,izn2,ikc2,ikd2,ld,izn1,ikc1,
        ikd1,izrez,kcrez,kdrez,izsum,izb,icb,idb;
    fp9=fopen("promprc","r");
    fscanf(fp9,"%d",&n);
    fprintf(fp5,"\nпроверка решения,матрица системы\n");
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            fprintf(fp5,"\nэлемент c(%d,%d)\n",i,j);
            fscanf(fp9,"%d%d%d",&rabl,&ncz,&ndz);
            znak[(i-1l)*n+j]=rabl;
            for(l=1;l<=ncz;l++)
            {
                fscanf(fp9,"%d",&rabl);
                ac[((i-1l)*n+j-1)*k50+l]=rabl;
            }
            for(l=1;l<=ndz;l++)
            {
                fscanf(fp9,"%d",&rabl);
                ad[((i-1l)*n+j-1)*k50+l]=rabl;
            }
            fprintf(fp5,"\nзнак числа=%2d\n", znak[(i-1l)*n+j]);
            fprintf(fp5,"\nколичество разрядов в целой=%3d,в дробной=%3d\n",
                ncz,ndz);
            fprintf(fp5,"\nцелая часть числа\n");
            for(ld=1;ld<=ncz;ld++)
            fprintf(fp5,"%1d", ac[((i-1l)*n+j-1)*k50+ld]);
            fprintf(fp5,"\ndробная часть числа\n");
            for(ld=1;ld<=ndz;ld++)
            fprintf(fp5,"%1d", ad[((i-1l)*n+j-1)*k50+ld]);
            fprintf(fp5,"\n");
            nc[(i-1l)*n+j]=ncz;
            nd[(i-1l)*n+j]=ndz;
        }
    }
//    fprintf(fp6,"\n проwg posle wiw matr\n");
//    if(n==3) return;
    fprintf(fp5,"\n свободные члены\n");
    for(i=1;i<=n;i++)
    {
        fprintf(fp5,"\n i=%3d\n",i);
        fscanf(fp9,"%d%d%d",&znp[i],&nbcz,&nbdz);
        for(l=1;l<=nbcz;l++)

```

```

{
fscanf(fp9,"%d",&rabl);
bc[(i-1l)*k50+1]=rabl;
}
for(l=1;l<=nbdz;l++)
{
fscanf(fp9,"%d",&rabl);
bd[(i-1l)*k50+1]=rabl;
}
fprintf(fp5,"\nзнак свободного члена %3d\n",znp[i]);
fprintf(fp5,"\nколичество разрядов в целой части=%3d,в дробной=%3d\n",
nbcz,nbdz);
fprintf(fp5,"\nцелая часть\n");
for(ld=1;ld<=nbcz;ld++)
fprintf(fp5,"%1d",bc[(i-1l)*k50+ld]);
fprintf(fp5,"\ndробная часть\n");
for(ld=1;ld<=nbdz;ld++)
fprintf(fp5,"%1d",bd[(i-1l)*k50+ld]);
fprintf(fp5,"\n");
nbc[i]=nbcz;
nbd[i]=nbdz;
}
fprintf(fp5,"\n неизвестные системы\n");
for(i=1;i<=n;i++)
{
fprintf(fp5,"\nномер неизвестного-%3d\n",i);
nbcz=nxc[i];
nbdz=nxd[i];
fprintf(fp5,"\nзнак неизвестного %2d\n",znx[i]);
fprintf(fp5,"количество разрядов в целой части=%3d,в дробной=%3d\n",
nbcz,nbdz);
fprintf(fp5,"\nцелая часть\n");
for(ld=1;ld<=nbcz;ld++)
fprintf(fp5,"%1d",xc[(i-1l)*k50+ld]);
fprintf(fp5,"\ndробная часть\n");
for(ld=1;ld<=nbdz;ld++)
fprintf(fp5,"%1d",xd[(i-1l)*k50+ld]);
fprintf(fp5,"\n");
}
// fprintf(fp5,"\n peredras prowg\n");

// if(n==3) return;
for(i=1;i<=n;i++)
{
//c
sum=0
izsum=1;
isc=1;
isd=1;
icsum[1]=0;
idsum[1]=0;
for(j=1;j<=n;j++)
{
izn2=znx[j];
ikc2=nxc[j];
ikd2=nxd[j];
for(ld=1;ld<=ikc2;ld++)
ic2[ld]=xc[(j-1l)*k50+ld];
for(ld=1;ld<=ikd2;ld++)
id2[ld]=xd[(j-1l)*k50+ld];
izn1=znak[(i-1l)*n+j];
ikc1=nc[(i-1l)*n+j];

ikd1=nd[(i-1l)*n+j];

```

```

for (ld=1;ld<=ikc1;ld++)
ic1[ld]=ac[ ((i-1l)*n+j-1)*k50+ld];
for (ld=1;ld<=ikd1;ld++)
id1[ld]=ad[ ((i-1l)*n+j-1)*k50+ld];

ymnzn(izn1,izn2,&izrez,ikc1,ikd1,ic1,id1,ikc2,ikd2,ic2,id2,
&kcrez,&kdrez,icrez,idrez,krazr);
izn2=izrez;
ikc2=kcrez;
ikd2=kdrez;
for (ld=1;ld<=kcrez;ld++)
m323: ic2[ld]=icrez[ld];
for (ld=1;ld<=kdrez;ld++)
m423: id2[ld]=idrez[ld];
slzn(izsum,izn2,&izrez,isc,isd,icsum,idsum,ikc2,ikd2,
ic2,id2,&kcrez,&kdrez,icrez,idrez,krazr);

izsum=izrez;
isc=kcrez;
isd=kdrez;
for (ld=1;ld<=kcrez;ld++)
m555: icsum[ld]=icrez[ld];
for (ld=1;ld<=kdrez;ld++)
m666: idsum[ld]=idrez[ld];
}
izb=znz[i];
icb=ncb[i];
idb=ndb[i];
for (ld=1;ld<=icb;ld++)
m815: ic1[ld]=bc[ (i-1l)*k50+ld];
for (ld=1;ld<=idb;ld++)
m816: id1[ld]=bd[ (i-1l)*k50+ld];
wiczn(izsum,izb,&izrez,isc,isd,icsum,idsum,icb,idb,ic1,id1,
&kcrez,&kdrez,icrez,idrez,krazr);
fprintf(fp5,"\n");
fprintf(fp5,"\n уравнение-%3d\n",i);
fprintf(fp5,"\n");
fprintf(fp5,"\nправая часть,исход\n");
fprintf(fp5,"\nзнак=%2d\n",izb);
fprintf(fp5,"\nдлина целой=%3d длина дробной=%3d\n", icb,idb );
fprintf(fp5,"\n целая часть=\n");
for (ld=1;ld<=icb;ld++)
fprintf(fp5,"%1d", ic1[ld]);
fprintf(fp5,"\n дробная часть=\n");
for (ld=1;ld<=idb;ld++)
fprintf(fp5,"%1d", id1[ld]);
fprintf(fp5,"\n");

fprintf(fp5,"\n правая часть,счет\n");
fprintf(fp5,"\nзнак=%2d\n",izsum);
fprintf(fp5,"\nдлина целой=%3d длина дробной=%3d\n", isc,isd);
fprintf(fp5,"\nцелая часть\n");
for (ld=1;ld<=isc;ld++)
fprintf(fp5,"%1d", icsum[ld]);
fprintf(fp5,"\nдробная часть\n");
for (ld=1;ld<=isd;ld++)
fprintf(fp5,"%1d", idsum[ld]);
fprintf(fp5,"\n");
fprintf(fp5,"\n разница\n");
fprintf(fp5,"\nзнак=%2d\n",izrez);
fprintf(fp5,"\nдлина целой=%3d длина дробной=%3d\n", kcrez,kdrez);
fprintf(fp5,"\nцелая часть\n");
for (ld=1;ld<=kcrez;ld++)

```

```

    fprintf(fp5,"%ld", icrez[ld]);
    fprintf(fp5,"\ndробная часть\n");
    for(ld=1;ld<=kdrez;ld++)
    fprintf(fp5,"%ld", idrez[ld]);
    fprintf(fp5,"\n");
    }
    fclose(fp9);
//    fclose(fp5);
    return;
    }
#include<stdio.h>
#include<math.h>
#include<alloc.h>

    void nimax(int i,int j,int n,
    int *nc ,int *nd , int *znak ,
    int *ac ,int *ad ,
    int *nbc ,int *nbd , int *znp ,
    int *bc ,int *bd ,
        int *mnp, int krazr,int k50)
    {
    int xmaxc[400],xmaxd[400] ,tc[400],td[400],ncel,
    ndr,zmax,kcmax,kdmax,ld,imax,jmax,ipoi,jpoi,ncr,
    kmin,jper,ktc,ktd,znt,iznr,nlc,nld;
//с-нахождение максимального элемента
//с  xmax=dabs(a(i,j))
    ncel=nc[(i-1l)*n+j];
    ndr=nd[(i-1l)*n+j];
    zmax=1;
    kcmax=ncel;
    kdmax=ndr;
    for(ld=1;ld<=ncel;ld++)
    xmaxc[ld]=ac[((i-1l)*n+j-1)*k50+ld];
    for(ld=1;ld<=ndr;ld++)
    xmaxd[ld]=ad[((i-1l)*n+j-1)*k50+ld];

    imax=i;
    jmax=j;
    for(ipoi=i;ipoi<=n;ipoi++)

    {
    for(jpoi=j;jpoi<=n;jpoi++)
    {
    ncr=nc[(ipoi-1l)*n+jpoi];
    ndr=nd[(ipoi-1l)*n+jpoi];
    if(ncr<kcmax) goto m38250;
    if(ncr>kcmax) goto m41003;
    for(ld=1;ld<=ncr;ld++)
    {
    if(ac[((ipoi-1l)*n+jpoi-1)*k50+ld]>xmaxc[ld]) goto m41003;
    if(ac[((ipoi-1l)*n+jpoi-1)*k50+ld]<xmaxc[ld]) goto m38250;
    }
    kmin=kdmax;
    if(kmin>ndr) kmin=ndr;
    for(ld=1;ld<=kmin;ld++)
    {
    if(ad[((ipoi-1l)*n+jpoi-1)*k50+ld]>xmaxd[ld]) goto m41003;
    if(ad[((ipoi-1l)*n+jpoi-1)*k50+ld]<xmaxd[ld]) goto m38250;
    }
    if(kdmax>=ndr) goto m38250;
m41003:;

```

```

imax=ipoi;
jmax=jpoi;
kcmx=ncr;
kdmx=ndr;
for (ld=1;ld<=ncr;ld++)
xmaxc[ld]=ac[ ((ipoi-1l)*n+jpoi-1)*k50+ld];
for (ld=1;ld<=ndr;ld++)
xmaxd[ld]=ad[ ((ipoi-1l)*n+jpoi-1)*k50+ld];
m38250:;
}}
//с-пересылка строки i со строкой imax
for (jper=j; jper<=n; jper++)
{
//c t=a(i, jper)
//c a(i, jper)=a(imax, jper)
//c a(imax, jper)=t
ktc=nc[ (i-1l)*n+jper];
ktd=nd[ (i-1l)*n+jper];
for (ld=1;ld<=ktd;ld++)
tc[ld]=ac[ ((i-1l)*n+jper-1)*k50+ld];
for (ld=1;ld<=ktd;ld++)
td[ld]=ad[ ((i-1l)*n+jper-1)*k50+ld];
znt=znak[ (i-1l)*n+jper];
ncr=nc[ (imax-1l)*n+jper];
ndr=nd[ (imax-1l)*n+jper];
for (ld=1;ld<=ncr;ld++)
ac[ ((i-1l)*n+jper-1)*k50+ld]=ac[ ((imax-1l)*n+jper-1)*k50+ld];
for (ld=1;ld<=ndr;ld++)
ad[ ((i-1l)*n+jper-1)*k50+ld]=ad[ ((imax-1l)*n+jper-1)*k50+ld];
znak[ (i-1l)*n+jper]=znak[ (imax-1l)*n+jper];
nc[ (i-1l)*n+jper]=ncr;
nd[ (i-1l)*n+jper]=ndr;
for (ld=1;ld<=ktd;ld++)
ac[ ((imax-1l)*n+jper-1)*k50+ld]=tc[ld];
for (ld=1;ld<=ktd;ld++)
ad[ ((imax-1l)*n+jper-1)*k50+ld]=td[ld];
znak[ (imax-1l)*n+jper]=znt;
nc[ (imax-1l)*n+jper]=ktd;
nd[ (imax-1l)*n+jper]=ktd;
}
ncr=ncr[i];
ndr=ndr[i];
iznr=znp[i];
for (ld=1;ld<=ncr;ld++)
tc[ld]=bc[ (i-1l)*k50+ld];
for (ld=1;ld<=ndr;ld++)
td[ld]=bd[ (i-1l)*k50+ld];
ncr[i]=ncr[imax];
ndr[i]=ndr[imax];
znp[i]=znp[imax];
n1c=ncr[imax];
n1d=ndr[imax];
for (ld=1;ld<=n1c;ld++)
bc[ (i-1l)*k50+ld]=bc[ (imax-1l)*k50+ld];
for (ld=1;ld<=n1d;ld++)
bd[ (i-1l)*k50+ld]=bd[ (imax-1l)*k50+ld];
ncr[imax]=ncr;
ndr[imax]=ndr;
znp[imax]=znp;
for (ld=1;ld<=ncr;ld++)
bc[ (imax-1l)*k50+ld]=tc[ld];
for (ld=1;ld<=ndr;ld++)
bd[ (imax-1l)*k50+ld]=td[ld];

```

```

//c t=b(i)
//c b(i)=b(imax)
//c b(imax)=t
//c-пересылка столбца j со столбцом jmax
    for(ipoi=1;ipoi<=n;ipoi++)
    {
//c t=a(ipoi,j)
//c a(ipoi,j)=a(ipoi,jmax)
//c a(ipoi,jmax)=t
ncr=nc[(ipoi-1l)*n+j];
ndr=nd[(ipoi-1l)*n+j];
iznr=znak[(ipoi-1l)*n+j];
for(ld=1;ld<=ncr;ld++)
tc[ld]=ac[((ipoi-1l)*n+j-1)*k50+ld];
for(ld=1;ld<=ndr;ld++)
td[ld]=ad[((ipoi-1l)*n+j-1)*k50+ld];
nc[(ipoi-1l)*n+j]=nc[(ipoi-1l)*n+jmax];
nd[(ipoi-1l)*n+j]=nd[(ipoi-1l)*n+jmax];
znak[(ipoi-1l)*n+j]=znak[(ipoi-1l)*n+jmax];
n1c=nc[(ipoi-1l)*n+jmax];
n1d=nd[(ipoi-1l)*n+jmax];
for(ld=1;ld<=n1c;ld++)
ac[((ipoi-1l)*n+j-1)*k50+ld]=ac[((ipoi-1l)*n+jmax-1)*k50+ld];
for(ld=1;ld<=n1d;ld++)
ad[((ipoi-1l)*n+j-1)*k50+ld]=ad[((ipoi-1l)*n+jmax-1)*k50+ld];
nc[(ipoi-1l)*n+jmax]=ncr;
nd[(ipoi-1l)*n+jmax]=ndr;
znak[(ipoi-1l)*n+jmax]=iznr;
for(ld=1;ld<=ncr;ld++)
ac[((ipoi-1l)*n+jmax-1)*k50+ld]=tc[ld];
for(ld=1;ld<=ndr;ld++)
ad[((ipoi-1l)*n+jmax-1)*k50+ld]=td[ld];
}
//c-пересылка в массиве номеров mpper столбцов
jper=mpper[j];
mpper[j]=mpper[jmax];
mpper[jmax]=jper;
return;
}
#include<stdio.h>
#include<math.h>
#include<alloc.h>

```

```

/*
с-обратный ход
с-метод Гаусса решения систем с выбором главного элемента, длинная арифметика
с-ac(i,j,l)-целая часть элемента i,j матрицы системы
с-ad(I,j,l)-дробная часть элемента I,j матрицы системы
с-nc(I,j)-количество разрядов целой части элемента I,j
с-nd(I,j)-количество разрядов дробной части элемента I,j
с-znak(I,j)-знак элемента I,j
с-bc(I,l)-целая часть правой части i-ого уравнения
с-bd(I,l)-дробная часть правой части i-ого уравнения
с-nbc(i)-длина целой части правой части i-го уравнения
с-nbd(i)-длина дробной части правой части i-го уравнения
с-zpr(i)-знак правой части i-ого уравнения
с-хс(I,l)-целая часть i-го неизвестного
с-хд(I,l)-дробная часть i-го неизвестного
с-пхс(i)-длина целой части i-ого неизвестного
с-пхд(i)-длина дробной части i-ого неизвестного
с-zпх(i)-знак i-ой неизвестной */

```

```

void obrdl(int n,int *znak ,
int *nc ,int *nd ,

int *ac ,int *ad ,
int *znp ,int *nbc ,int *nbd ,
int *bc ,int *bd ,
int *znx ,int *nxc ,int *nxd ,
int *xc ,int *xd ,
int *pigau,int *mper,int anc[400],
int andr[400],int krazr,int k50)
{
extern FILE *fp6;

extern void ymnzn(int izr,int izl,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int ncl,int ndl,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void delzn(int izr,int izt,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int nct,int ndt,int *tc,int *td,
int *pkcrez,int *pkdrez,int *rezc,int *rezd, int krazr);

extern void slzn(int izsum,int izr,int *pizrez,
int ncsun,int ndsum,int *icsun,int *idsum,
int ncr,int ndr,int *rc,int *rd,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

extern void wiczn(int izr,int izsum,int *pizrez,
int ncr,int ndr,int *rc,int *rd,
int ncsun,int ndsum,int *icsun,int *idsum,
int *pkcrez,int *pkdrez,int *rezc,int *rezd,int krazr);

int rezc[400],rezd[400],i,*pizrez,*pkcrez,*pkdrez,igau,
icsun[400],idsum[400],tc[400],td[400];
int rc[400],rd[400],nis,nct,ndt,izt,ld,nt,ncr,ndr,izr,
kcrez,kdrez,izrez,ncsun,ndsum,izsum,ncl,ndl,izl,j,nlc,nld;
nis=n;
nct=nc[(n-1)*n+n];
ndt=nd[(n-1)*n+n];
izt=znak[(n-1)*n+n];
for(ld=1;ld<=nct;ld++)
tc[ld]=ac[((n-1)*n+n-1)*k50+ld];
for(ld=1;ld<=ndt;ld++)
td[ld]=ad[((n-1)*n+n-1)*k50+ld];
for(ld=1;ld<=nct;ld++)
if(tc[ld]!=anc[ld]) goto m82;
for(ld=1;ld<=ndt;ld++)
if(td[ld]!=andr[ld]) goto m82;
fprintf(fp6," матрица вырожденная\n");
igau=0;
*pigau=0;
return;
m82:;
nt=n;
//c x(n)=b(n)/a(n,n)
ncr=nbc[n];
ndr=nbnd[n];
izr=znp[n];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(n-1)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=bd[(n-1)*k50+ld];

```

```

delzn(izr, izt, &izrez, ncr, ndr, rc, rd, nct, ndt, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
nxc[n]=kcrez;
nxd[n]=kdrez;
znx[n]=izrez;
for(ld=1;ld<=kcrez;ld++)
xc[(n-11)*k50+ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
xd[(n-11)*k50+ld]=rezd[ld];
m85: n=n-1;
if(n<1) goto m83;
//c sum=0
ncsum=1;
ndsum=1;
izsum=1;
for(ld=1;ld<=ncsum;ld++)
{
icsum[ld]=0;
idsun[ld]=0;
}
j=n+1;
m84:;
ncr=nc[(n-11)*nt+j];
ndr=nd[(n-11)*nt+j];
izr=znak[(n-11)*nt+j];
for(ld=1;ld<=ncr;ld++)
rc[ld]=ac[(n-11)*nt+j-1)*k50+ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=ad[(n-11)*nt+j-1)*k50+ld];
nc1=nxc[j];
nd1=nxd[j];
iz1=znx[j];
for(ld=1;ld<=nc1;ld++)
tc[ld]=xc[(j-11)*k50+ld];
for(ld=1;ld<=nd1;ld++)
td[ld]=xd[(j-11)*k50+ld];
ymnzn(izr, iz1, &izrez, ncr, ndr, rc, rd, nc1, nd1, tc, td,
&kcrez, &kdrez, rezc, rezd, krazr);
ncr=kcrez;
ndr=kdrez;
izr=izrez;
for(ld=1;ld<=ncr;ld++)
rc[ld]=rezc[ld];
for(ld=1;ld<=ndr;ld++)
rd[ld]=rezd[ld];
slzn(izsum, izr, &izrez, ncsum, ndsum, icsum, idsum, ncr, ndr,
rc, rd, &kcrez, &kdrez, rezc, rezd, krazr);
ncsum=kcrez;
ndsum=kdrez;
izsum=izrez;
for(ld=1;ld<=kcrez;ld++)
icsum[ld]=rezc[ld];
for(ld=1;ld<=kdrez;ld++)
idsun[ld]=rezd[ld];
//c84 sum=sum+a(n,j)*x(j)
j=j+1;
if(j<=nt) goto m84;
//c x(n)=b(n)-sum
ncr=nc[n];
ndr=nd[n];
izr=znx[n];
for(ld=1;ld<=ncr;ld++)
rc[ld]=bc[(n-11)*k50+ld];

```

```

    for (ld=1;ld<=ndr;ld++)
        rd[ld]=bd[(n-1l)*k50+ld];
    wiczn(izr, izsum, &izrez, ncr, ndr, rc, rd, ncsun, ndsum, icsun,
    idsum, &kcrez, &kdrez, rezc, rezd, krazr);
    nxc[n]=kcrez;
    nxd[n]=kdrez;
    znx[n]=izrez;
    for (ld=1;ld<=kcrez;ld++)
        xc[(n-1l)*k50+ld]=rezc[ld];
    for (ld=1;ld<=kdrez;ld++)
        xd[(n-1l)*k50+ld]=rezd[ld];
    goto m85;
m83:;
    igau=1;
    *pigau=igau;
    //с-расстановка значений в векторе решения X
    for (i=1;i<=nt;i++)
        {
        j=mpcr[i];
    //с b(j)=x(i)
        nbc[j]=nxc[i];
        nbd[j]=nxd[i];
        znp[j]=znx[i];
        nlc=nxc[i];
        nld=nxd[i];
        for (ld=1;ld<=nlc;ld++)
            bc[(j-1l)*k50+ld]=xc[(i-1l)*k50+ld];
        for (ld=1;ld<=nld;ld++)
            bd[(j-1l)*k50+ld]=xd[(i-1l)*k50+ld];
        }
    for (i=1;i<=nt;i++)
        {
    //с x(i)=b(i)
        nxc[i]=nbc[i];
        nxd[i]=nbd[i];
        znx[i]=znp[i];
        nlc=nbc[i];
        nld=nbd[i];
        for (ld=1;ld<=nlc;ld++)
            xc[(i-1l)*k50+ld]=bc[(i-1l)*k50+ld];
        for (ld=1;ld<=nld;ld++)
            xd[(i-1l)*k50+ld]=bd[(i-1l)*k50+ld];
        }
        n=nis;
        return;
    }

#include<stdio.h>
#include<math.h>
#include<alloc.h>

//с-длинное деление с учетом знаков
void delzn(int za,int zb,int *pzc,
    int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr)
    {
    // extern FILE *fp6;
    extern void deld(int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr);

```

```

int c[400],cd[400],ni,ndi,mi,mdi,kcdel,kddel,
ar[400],adr[400],br[400],bdr[400],zais,zbis,i,
ctot[400],cdtot[400],ld,
rezc[400],rezdr[400],ce[400],ced[400];
zais=za;
zbis=zb;
for(i=1;i<=n;i++)
ar[i]=a[i];
for(i=1;i<=nd;i++)
adr[i]=ad[i];
for(i=1;i<=m;i++)
br[i]=b[i];
for(i=1;i<=md;i++)
bdr[i]=bd[i];
ni=n;
ndi=nd;
mi=m;
mdi=md;
*psc=za*zb;
/*      fprintf(fp6,"делимое delzn\n");
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n", n,nd);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=n;ld++)
fprintf(fp6,"%ld", a[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=nd;ld++)
fprintf(fp6,"%ld", ad[ld]);
fprintf(fp6,"\n делитель \n");
//      fprintf(fp6,"знак=%d\n",iznt);
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n",m,md);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=m;ld++)
fprintf(fp6,"%ld", b[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=md;ld++)
fprintf(fp6,"%ld", bd[ld]);
fprintf(fp6,"\n"); */

deld(n,nd,a,ad,m,md,b,bd,&kcdel,&kddel,cdel,ddel,krazr);
/*      fprintf(fp6,"частное delzn\n");
fprintf(fp6,"длина целой части=%d длина дробной части=%d\n",kcdel,kddel);
fprintf(fp6,"\n целая часть\n");
for(ld=1;ld<=kcdel;ld++)
fprintf(fp6,"%ld", cdel[ld]);
fprintf(fp6,"\n дробная часть\n");
for(ld=1;ld<=kddel;ld++)
fprintf(fp6,"%ld", ddel[ld]);
fprintf(fp6,"\n"); */
n=ni;
nd=ndi;
m=mi;
md=mdi;
for(i=1;i<=n;i++)
a[i]=ar[i];
for(i=1;i<=nd;i++)
ad[i]=adr[i];
for(i=1;i<=m;i++)
b[i]=br[i];
for(i=1;i<=md;i++)
bd[i]=bdr[i];
za=zais;
zb=zbis;

```

```

    *pkcdel=kcdel;
    *pkddel=kddel;
    return;
}
//deld
void deld(int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcdel,int *pkddel,int *cdel,int *ddel,int krazr)
{

extern void wich(int nat,int kad,int *a,int *ad,
int mat,int kbd,int *b,int *bd,
int *pkc,int *pkcd,int *c,int *cd,int krazr);

extern void sl(int kctot,int kdto,int *ctot,int *cdtot,
int kedi,int kedid,int *ce,int *ced,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

//с-длинное деление
int c[400],cd[400],kcdel,kddel,
ctot[400],cdtot[400],*pkc,*pkcd,*pkcel,*pkdro,
rezc[400],rezdr[400],ce[400],ced[400];
int i,max,in,it,ik,il,j,ill,jp,kctot,kdto,
kedi,kedid,kad,kbd,nat,mat,isd,kcel,kdro,kc,kcd;

for(i=1;i<=399;i++)
{

    cdel[i]=0;
    ddel[i]=0;
}
//с-формирование двух массивов для деления
max=nd;
if(max<md) max=md;
if(md==max) goto m78;
in=md+1;
for(i=in;i<=max;i++)
bd[i]=0;
goto m81;
m78:;
in=nd+1;
for(i=in;i<=max;i++)
ad[i]=0;
m81:;
it=0;
in=n+1;
ik=n+max;
for(i=in;i<=ik;i++)
{

    it=it+1;

    a[i]=ad[it];
}

il=0;
for(j=1;j<=ik;j++)
{

    if(a[j]!=0) goto m5672;
    il=il+1;
}
kcdel=0;

```

```

        kddel=0;
        goto m1231;
//      exit(0);
m5672:;
        it=0;
        for(jp=j;jp<=ik;jp++)
        {

            it=it+1;
            a[it]=a[jp];
        }

        it=0;
        in=m+1;
        ik=m+max;

        for(i=in;i<=ik;i++)
        {

            it=it+1;
            b[i]=bd[it];
        }

        ill=0;
        for(j=1;j<=ik;j++)
        {

            if(b[j]!=0) goto m15672;
            ill=ill+1;
        }
        kcdel=0;

        kddel=0;
        goto m1231;
//      exit(0);
m15672:;
        it=0;
        for(jp=j;jp<=ik;jp++)
        {

            it=it+1;
            b[it]=b[jp];
        }

//c          if(n.lt.m) goto 675
//c          if(n.eq.m) goto 676
        for(i=1;i<=399;i++)
        {
            ctot[i]=0;
            cdtot[i]=0;
        }
        kctot=1;
        kdto=1;

        ce[1]=1;
        kedi=1;
        ced[1]=0;
        kedid=1;
        kad=1;
        kbd=1;
        ad[1]=0;
        bd[1]=0;

```

```

    nat=n+max-1;
    mat=m+max-1;
    isd=0;
    kcel=1;
    kdrc=1;
    rezc[1]=0;
    rezdr[1]=0;
m2010:;
    if(nat<mat) goto m675;
    if(nat==mat) goto m676;
    goto m677;
m675:;
    if(isd!=0) goto m5621;
    for(j=1;j<=kcel;j++)
        cdel[j]=rezc[j];
    for(j=1;j<=kdrc;j++)
        ddel[j]=rezdr[j];
    rezc[1]=0;
    kcdel=kcel;
    kdrcdel=kdrc;
    if(nat==0) goto m1231;
m5621:;
    if(isd==0) goto m8721;
    ddel[isd]=rezc[1];
    rezc[1]=0;
    kdrcdel=isd;
m8721:;
    if(nat==0) goto m1231;
    for(j=1;j<=nat;j++)
    {
        if(a[j]!=0) goto m9782;
    }

    goto m1231;
m9782:;
    nat=nat+1;
    isd=isd+1;

    if(isd>krazr) goto m1231;
    a[nat]=0;
    for(i=1;i<=399;i++)
    {
        ctot[i]=0;
        cdtot[i]=0;
    }
    kctot=1;
    kdrc=1;
    goto m2010;
m676:;
    for(j=1;j<=nat;j++)
    {
        if(a[j]>b[j]) goto m677;
        if(a[j]<b[j]) goto m675;
    }
m677:;
    wich(nat,kad,a,ad,mat,kbd,b,bd,&kc,&kcd,c,cd,krazr);
    sl(kctot,kdrc,ctot,cdtot,kedi,kedid,ce,ced,
    &kcel,&kdrc,rezc,rezdr,krazr);
    kctot=kcel;
    kdrc=kdrc;
    for(i=1;i<=kcel;i++)
        ctot[i]=rezc[i];

```

```

    for(i=1;i<=kdro;i++)
    cdtot[i]=rezdr[i];
    nat=kc;
    kad=kcd;
    if(nat==0) a[1]=0;
    for(j=1;j<=nat;j++)
    a[j]=c[j];
    for(j=1;j<=kad;j++)
    ad[j]=cd[j];
    goto m2010;

m1231:;
    if(kcdel!=0) goto m971;
    kcdel=1;
    cdel[1]=0;
m971:;
    if(kddel!=0) goto m972;
    kddel=1;

    ddel[1]=0;
m972:;
    if(kddel>krazr) kddel=krazr;
    *pkcdel=kcdel;
    *pkddel=kddel;
    return;
}
#include<stdio.h>
#include<math.h>
#include<alloc.h>

void ymnzn(int za,int zb,int *pzc,int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr)
{

extern void ymn(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr);

//с-длинное умножение с учетом знаков сомножителей
int c[400],cd[400],kcel,kdro,zc,
    csl[400],csld[400],
    rezc[400],rezdr[400],
    ai[400],adi[400],bi[400],bdi[400];
int zais,zbis,i,ni,ndi,mi,mdi;

zais=za;
zbis=zb;
for(i=1;i<=n;i++)
ai[i]=a[i];
for(i=1;i<=nd;i++)

adi[i]=ad[i];
for(i=1;i<=m;i++)
bi[i]=b[i];
for(i=1;i<=md;i++)
bdi[i]=bd[i];
ni=n;
ndi=nd;
mi=m;
mdi=md;

```

```

*psc=za*zb;

ymn(n,nd,a,ad,m,md,b,bd,pkcel,pkdro,ctot,cdtot,krazr);

n=ni;
nd=ndi;
m=mi;
md=mdi;
for(i=1;i<=n;i++)
a[i]=ai[i];
for(i=1;i<=nd;i++)
ad[i]=adi[i];
for(i=1;i<=m;i++)
b[i]=bi[i];
for(i=1;i<=md;i++)
bd[i]=bdi[i];
za=zais;
zb=zbis;

return;
}
//с-длинное умножение ymn

void ymn(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *ctot,int *cdtot,int krazr)
{

extern void sl(int icel,int idr,int *csl,int *csld,
int ntot,int kdtot,int *ctot,int *cdtot,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

int c[400],cd[400],i,it,in,ik,ntot,kdtot,im,p,nt,mt,jp,pr,il,os,
kraz,kcd,icel,iab,idr,j,il,kcel,kdro,
csl[400],csld[400],
rezc[400],rezdr[400];
for(i=1;i<=399;i++)
{
ctot[i]=0;
cdtot[i]=0;
}
//с-формирование двух массивов для умножения
it=0;
in=n+1;
ik=n+nd;
for(i=in;i<=ik;i++)
{
it=it+1;
a[i]=ad[it];
}
it=0;
in=m+1;
ik=m+md;
for(i=in;i<=ik;i++)
{
it=it+1;
b[i]=bd[it];
}
ntot=n+m;
kdtot=nd+md;
for(i=1;i<=ntot;i++)

```

```

ctot[i]=0;
for (i=1;i<=kdtot;i++)
cdtot[i]=0;

//с-умножение
i=0;
im=0;
p=0;
nt=n+nd;
mt=m+md;
m801:;
p=0;
for (jp=1;jp<=399;jp++)
c[jp]=0;
i=0;
nt=n+nd;
im=im+1;
m701:;
pr=a[nt]*b[mt]+p;
i1=pr/10;
os=pr-i1*10;
i=i+1;
c[i]=os;
p=i1;
nt=nt-1;
if (nt>=1) goto m701;
if (p==0) goto m1701;
i=i+1;
c[i]=p;
p=0;
m1701:;
kraz=md-im+1;
kcd=nd+kraz;
icel=i-kcd;
if (icel<i) goto m1921;
iab=kcd;
if (kcd<0) iab=-kcd;
it=0;
for (jp=i;jp>=1;jp=jp-1)
{
it=it+1;
csl[it]=c[jp];
}
if (iab==0) goto m1941;
for (jp=iab;jp>=1;jp=jp-1)
{
it=it+1;
csl[it]=0;
}
m1941:;
idr=1;
csld[1]=0;
goto m8722;
m1921:;
it=0;
in=i;
ik=kcd+1;
if (in==ik) goto m961;
if (in<ik) goto m962;
for (j=in;j>=ik;j=j-1)
{
it=it+1;
csl[it]=c[j];
}

```

```

    }
    icel=i-kcd;
    goto m963;
m961:;
    icel=1;
    csl[1]=c[in];
    goto m963;
m962:;
    icel=1;
    csl[1]=0;
m963:;
    in=kcd;
    it=0;
    for (j=in; j>=1; j=j-1)
    {
        it=it+1;
        csld[it]=c[j];
    }
    idr=kcd;
    if(idr>0) goto m8722;
    idr=1;
    csld[1]=0;
m8722:;

    sl(icel, idr, csl, csld, ntot, kdtot, ctot, cdtot, &kcel, &kdro,
    rezc, rezdr, krazr);
    if(kcel!=0) goto m8723;
    kcel=1;
    rezc[1]=0;
m8723:;
    if(kdro!=0) goto m8724;
    kdro=1;
    rezdr[1]=0;
m8724:;
    for (i=1; i<=kcel; i++)
    ctot[i]=rezc[i];
    for (i=1; i<=kdro; i++)
    cdtot[i]=rezdr[i];
    ntot=kcel;
    kdtot=kdro;
    mt=mt-1;
    if(mt>=1) goto m801;
    it=0;
    for (j=1; j<=kcel; j++)
    {
        if(ctot[j]!=0) goto m7767;
        it=it+1;
    }
m7767:;
    il=0;
    for (jp=j; jp<=kcel; jp++)
    {
        il=il+1;
        ctot[il]=ctot[jp];
    }
    kcel=kcel-it;
    if(kcel!=0) goto m3999;
    kcel=1;
    ctot[1]=0;
m3999:;
    if(kdro!=0) goto m4000;
    kdro=1;
    cdtot[1]=0;

```

```

m4000:;

    if(kdro>krazr) kdro=krazr;
    *pkcel=kcel;
    *pkdro=kdro;
    return;

}

#include<stdio.h>
#include<math.h>
#include<alloc.h>

//с-длинное сложение
void sl(int n,int nd,int *a,int *ad,
int m,int md,int *b,int *bd,
int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{
int c[401],cd[401],i,j,ir,max,in,kcel,kdro,
it,ik,p,nt,mt,rez,il,os;

    for(i=1;i<=399;i++)
    {
    rezc[i]=0;
    rezdr[i]=0;
    }
    if(n>=m) goto m897;
    for(i=1;i<=n;i++)
    c[i]=a[i];
    for(i=1;i<=nd;i++)
    cd[i]=ad[i];
    for(i=1;i<=m;i++)
    a[i]=b[i];
    for(i=1;i<=md;i++)
    ad[i]=bd[i];
    for(i=1;i<=n;i++)
    b[i]=c[i];
    for(i=1;i<=nd;i++)
    bd[i]=cd[i];
    ir=n;
    n=m;
    m=ir;
    ir=nd;
    nd=md;
    md=ir;
m897:;
//с-формирование двух массивов для сложения
    max=nd;
    if(max<md) max=md;
    if(md==max) goto m78;
    in=md+1;
    for(i=in;i<=max;i++)
    bd[i]=0;
    goto m81;
m78:;
    in=nd+1;
    for(i=in;i<=max;i++)
    ad[i]=0;
m81:;
    it=0;
    in=n+1;
    ik=n+max;
    for(i=in;i<=ik;i++)

```

```

    {
    it=it+1;
    a[i]=ad[it];
    }
    it=0;
    in=m+1;
    ik=m+max;
    for (i=in; i<=ik; i++)
    {
    it=it+1;
    b[i]=bd[it];
    }

//с-сложение
    i=0;
    p=0;
    nt=n+max;
    mt=m+max;
m7:;
    if (mt<1) rez=a[nt];
    if (mt>=1) rez=a[nt]+b[mt];
    rez=rez+p;
    il=rez/10;
    os=rez-il*10;
    i=i+1;
    c[i]=os;
//с
    if (i.ne.1) goto 1
    p=il;
    nt=nt-1;
    mt=mt-1;
    if ((p==0) && (mt<1)) goto m4;
    if ((nt<1) && (p!=0)) goto m5;
    if ((nt<1) && (p==0)) goto m4;
    goto m7;
m5:;
    i=i+1;
    c[i]=p;
    goto m67;
m4:;

m89:;
    if (nt<1) goto m67;
    i=i+1;
    c[i]=a[nt];
    nt=nt-1;
    goto m89;
m67:;
    ik=max+1;
    kcel=i-max;
    kdro=max;
    it=0;
    for (j=i; j>=ik; j=j-1)
    {
    it=it+1;
    rezc[it]=c[j];
    }
    it=0;
    for (j=max; j>=1; j=j-1)
    {
    it=it+1;
    rezdr[it]=c[j];
    }
    if (kcel!=0) goto m971;

```

```

    kcel=1;
    rezc[1]=0;
m971:;
    if(kdro!=0) goto m972;
    kdro=1;
    rezdr[1]=0;
m972:;
    if(kdro>krazr) kdro=krazr;
    *pkcel=kcel;
    *pkdro=kdro;

    return;
}

#include<stdio.h>
#include<math.h>
#include<alloc.h>

    void slzn(int za,int zb,int *pzc,
    int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{

    extern void sl(int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

    extern void wich(int n,int nd,int *a,int *ad,
    int m,int md,int *b,int *bd,
    int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

//с-длинное сложение с учетом знаков слагаемых

    int c[401],cd[401],kcel,kdro,
    ar[401],adr[401],zc,
    ai[401],adi[401],bi[401],bdi[401];
    int zais,zbis,i,ni,ndi,mi,mdi,ip,nmin,nr,ndr;
    zais=za;
    zbis=zb;
    for(i=1;i<=n;i++)
    ai[i]=a[i];
    for(i=1;i<=nd;i++)
    adi[i]=ad[i];
    for(i=1;i<=m;i++)
    bi[i]=b[i];
    for(i=1;i<=md;i++)
    bdi[i]=bd[i];
    ni=n;
    ndi=nd;
    mi=m;
    mdi=md;

    ip=za*zb;
    if(ip>0) goto m781;
    if(za==1) goto m782;
m1901:    if(m>n) goto m783;

    if(m==n) goto m784;
    wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
    *pzc=za;
    goto m785;
m784:;

```

```

for(i=1;i<=m;i++)
{
if(a[i]>b[i]) goto m787;
if(a[i]<b[i]) goto m788;
}
nmin=nd;
if(nmin>md) nmin=md;
for(i=1;i<=nmin;i++)
{

if(ad[i]>bd[i]) goto m792;
if(ad[i]<bd[i]) goto m793;
}
if(nd<=md) goto m795;
wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
*psc=za;
goto m785;
m795:;
wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
*psc=zb;
goto m785;
m793:;
wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
*psc=zb;
goto m785;
m792:;
wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
*psc=za;
goto m785;
m788:;
wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
*psc=zb;
goto m785;
m787:;
wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
*psc=za;
goto m785;
m783:;
wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
*psc=zb;
goto m785;
m782:;
for(i=1;i<=n;i++)
ar[i]=a[i];

for(i=1;i<=nd;i++)
adr[i]=ad[i];
nr=n;
ndr=nd;
for(i=1;i<=m;i++)
a[i]=b[i];
for(i=1;i<=md;i++)
ad[i]=bd[i];
n=m;
nd=md;
m=nr;
md=ndr;
for(i=1;i<=nr;i++)
b[i]=ar[i];
for(i=1;i<=ndr;i++)
bd[i]=adr[i];
zc=za;
za=zb;

```

```

        zb=zc;
        goto m1901;
m781:;
        sl(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=za;
m785:;
        n=ni;
        nd=ndi;
        m=mi;
        md=mdi;
        for(i=1;i<=n;i++)
            a[i]=ai[i];
        for(i=1;i<=nd;i++)
            ad[i]=adi[i];
        for(i=1;i<=m;i++)
            b[i]=bi[i];
        for(i=1;i<=md;i++)
            bd[i]=bdi[i];

        za=zais;
        zb=zbis;
        *pkcel=kcel;
        *pkdro=kdro;
        return;

    }

```

```

#include<stdio.h>
#include<math.h>
#include<alloc.h>

```

```

//с-длинное вычитание

```

```

        void wich(int n,int nd,int *a,int *ad,
            int m,int md,int *b,int *bd,
            int *pkcra,int *pkdra,int *cra,int *dra,int krazr)
        {
int c[401],cd[401],i,ir,max,jp,kcra,kdra,
in,it,ik,nt,mt,np,il,j,im;

        for(i=1;i<=399;i++)
        {
            cra[i]=0;
            dra[i]=0;
        }
        if(n>=m) goto m897;
        for(i=1;i<=n;i++)
            c[i]=a[i];
        for(i=1;i<=nd;i++)
            cd[i]=ad[i];
        for(i=1;i<=m;i++)
            a[i]=b[i];
        for(i=1;i<=md;i++)
            ad[i]=bd[i];
        for(i=1;i<=n;i++)
            b[i]=c[i];
        for(i=1;i<=nd;i++)
            bd[i]=cd[i];
        ir=n;
        n=m;
        m=ir;
        ir=nd;
        nd=md;
        md=ir;

```

```

m897:;

//с-формирование двух массивов для вычитания
max=nd;
if(max<md) max=md;
if(md==max) goto m78;
in=md+1;
for(i=in;i<=max;i++)
bd[i]=0;
goto m81;
m78:;
in=nd+1;
for(i=in;i<=max;i++)
ad[i]=0;
m81:;
it=0;
in=n+1;
ik=n+max;
for(i=in;i<=ik;i++)
{
it=it+1;
a[i]=ad[it];
}
it=0;
in=m+1;
ik=m+max;
for(i=in;i<=ik;i++)
{
it=it+1;
b[i]=bd[it];
}
i=0;
nt=n+max;
mt=m+max;
m10:;
if(a[nt]>=b[mt]) goto m1;
np=nt-1;
m3:;
if(np==0) goto m201;
if(a[np]!=0) goto m2;
np=np-1;
goto m3;
m2:;
a[np]=a[np]-1;
m7:;
np=np+1;

if(np==nt) goto m30;
a[np]=9;
goto m7;
m30:;
a[nt]=a[nt]+10;
goto m10;
m1:;
i=i+1;
c[i]=a[nt]-b[mt];
nt=nt-1;
mt=mt-1;
if(mt!=0) goto m10;
if(n==m) goto m781;
m561:;
i=i+1;
c[i]=a[nt];

```

```

        nt=nt-1;
        if(nt>=1) goto m561;
m781:;
        in=i;
        ik=max+1;
        in=max;
        goto m999;
m201:;
// write(5,*) 'из меньшего вычитаем большее,здесь нельзя!'
// exit(0);
return;
m999:;
        kcra=i-max;
        kdra=max;
        it=0;
        in=i;
        ik=max+1;
        for(j=i;j>=ik;j=j-1)
        {
            it=it+1;
            cra[it]=c[j];
        }
        im=0;
        for(j=1;j<=it;j++)
        {
            if(cra[j]!=0) goto m3122;
            im=im+1;
        }
        goto m973;
m3122:;
        il=0;
        for(jp=j;jp<=it;jp++)
        {
            il=il+1;
            cra[il]=cra[jp];
        }
m973:;
        kcra=kcra-im;
        it=0;
        in=max;
        for(j=max;j>=1;j=j-1)
        {
            it=it+1;
            dra[it]=c[j];
        }
        if(kcra!=0) goto m971;
        kcra=1;
        cra[1]=0;
m971:;
        if(kdra!=0) goto m972;
        kdra=1;
        dra[1]=0;
m972:;
        if(kdra>krazr) kdra=krazr;
        *pkcra=kcra;
        *pkdra=kdra;
        return;
    }

#include<stdio.h>
#include<math.h>

```

```

#include<alloc.h>

void wiczn(int za,int zb,int *pzc,
  int n,int nd,int *a,int *ad,
  int m,int md,int *b,int *bd,
  int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr)
{
  extern void sl(int n,int nd,int *a,int *ad,
  int m,int md,int *b,int *bd,
  int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

  extern void wich(int n,int nd,int *a,int *ad,
  int m,int md,int *b,int *bd,
  int *pkcel,int *pkdro,int *rezc,int *rezdr,int krazr);

  // с-длинное вычитание с учетом знаков уменьшаемого и вычитаемого
  int c[401],cd[401],kcel,kdro,zc,
  ar[401],adr[401],
  ai[401],adi[401],bi[401],bdi[401],zais,zbi,ip,
  zbis,i,ni,ndi,mi,mdi,nmin,nr,ndr;

  zais=za;
  zbis=zb;
  for(i=1;i<=n;i++)
  ai[i]=a[i];
  for(i=1;i<=nd;i++)
  adi[i]=ad[i];
  for(i=1;i<=m;i++)
  bi[i]=b[i];
  for(i=1;i<=md;i++)
  bdi[i]=bd[i];
  ni=n;
  ndi=nd;
  mi=m;
  mdi=md;
  zbi=zb;
  zb=zb*(-1);
  ip=za*zb;
  if(ip>0) goto m781;
  if(za==1) goto m782;
m1901: if(m>n) goto m783;
  if(m==n) goto m784;
  wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
  goto m785;
m784:;
  for(i=1;i<=m;i++)
  {
  if(a[i]>b[i]) goto m787;
  if(a[i]<b[i]) goto m788;
  }
  nmin=nd;
  if(nmin>md) nmin=md;
  for(i=1;i<=nmin;i++)
  {
  if(ad[i]>bd[i]) goto m792;
  if(ad[i]<bd[i]) goto m793;
  }
  if(nd<=md) goto m795;
  wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
  *pzc=za;
  goto m785;
m795:;

```

```

        wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=zb;
        goto m785;
m793:;
        wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);

        *pzc=zb;
        goto m785;
m792:;
        wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=za;
        goto m785;
m788:;
        wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=zb;
        goto m785;
m787:;
        wich(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=za;
        goto m785;
m783:;
        wich(m,md,b,bd,n,nd,a,ad,&kcel,&kdro,rezc,rezdr,krazr);
        *pzc=zb;
        goto m785;
m782:;
        for(i=1;i<=n;i++)
            ar[i]=a[i];
        for(i=1;i<=nd;i++)
            adr[i]=ad[i];
        nr=n;
        ndr=nd;
        for(i=1;i<=m;i++)
            a[i]=b[i];
        for(i=1;i<=md;i++)
            ad[i]=bd[i];
        n=m;
        nd=md;
        m=nr;
        md=ndr;
        for(i=1;i<=nr;i++)
            b[i]=ar[i];
        for(i=1;i<=ndr;i++)
            bd[i]=adr[i];
        zc=za;
        za=zb;
        zb=zc;
        goto m1901;
m781:;
        sl(n,nd,a,ad,m,md,b,bd,&kcel,&kdro,rezc,rezdr,krazr);

        *pzc=za;
m785:;
        zb=zbi;
        n=ni;
        nd=ndi;
        m=mi;
        md=mdi;
        for(i=1;i<=n;i++)
            a[i]=ai[i];
        for(i=1;i<=nd;i++)
            ad[i]=adi[i];
        for(i=1;i<=m;i++)
            b[i]=bi[i];

```

```
for (i=1; i<=md; i++)  
bd[i]=bdi[i];  
za=zais;  
zb=zbis;  
*pkcel=kcel;  
*pkdro=kdro;  
return;  
}
```



6.44680	.50583	.78657
6.62050	.16694	.92247
6.20676	.03241	.89217
3.09042	.06137	.33869
6.67043	.47979	.83051
5.37578	.66496	.51516
2.13248	.10636	.00174
2.94147	.05316	.30332
3.51719	.34145	.31882
4.67627	.83984	.16969
3.50219	.45961	.23042
3.65775	.24603	.40242
4.44886	.20027	.57985
7.39799	.67665	.84125
5.23350	.95769	.17486
2.81173	.42996	.00597
6.77843	.16619	.94343
3.96027	.16484	.49821
5.02686	.50572	.55040
5.89576	.75090	.55182
4.23219	.55276	.35021
4.47488	.12639	.60650
5.01392	.50148	.55037
7.41925	.78890	.78006
4.81356	.58027	.46106
3.40254	.54392	.11397
6.93649	.78855	.70552
3.76359	.13619	.46743
3.99500	.57888	.26646
7.18237	.38882	.93480
2.79783	.10943	.24337
5.56365	.47533	.66162
4.15539	.69173	.19582
9.23586	.94937	.93384
6.65467	.05870	.94761
3.55681	.15736	.41572
5.12934	.07807	.72692
4.85589	.65888	.41100
4.94883	.53607	.51742
2.48326	.27989	.02258
5.10257	.61640	.49524
2.97624	.42379	.08562
5.07909	.69959	.42693
5.58779	.26524	.74913
6.38553	.10758	.90335
5.03409	.45183	.58087
3.58119	.41872	.28450

### Набор name(имена параметров)

3

- (p 1)
- (p 2)
- (p 3)



i=3 j=1

1 1 9

целая часть  
7

дробная часть  
840720000

i=3 j=2

1 1 9

целая часть  
0

дробная часть  
644930000

i=3 j=3

1 1 9

целая часть  
0

дробная часть  
917970000

i=4 j=1

1 1 9

целая часть  
8

дробная часть  
370509999

i=4 j=2

1 1 9

целая часть  
0

дробная часть  
703460000

i=4 j=3

1 1 9

целая часть  
0

дробная часть  
959040000

i=5 j=1

1 1 9

целая часть

7

дробная часть

065220000

i=5 j=2

1 1 9

целая часть

0

дробная часть

423110000

i=5 j=3

1 1 9

целая часть

0

дробная часть

907290000

i=6 j=1

1 1 9

целая часть

6

дробная часть

005989999

i=6 j=2

1 1 9

целая часть

0

дробная часть

635770000

i=6 j=3

1 1 9

целая часть

0

дробная часть

649040000

i=7 j=1

1 1 9

целая часть

2

дробная часть  
727710000

i=7 j=2

1 1 9

целая часть  
0

дробная часть  
172300000

i=7 j=3

1 1 9

целая часть  
0

дробная часть  
192460000

i=8 j=1

1 1 9

целая часть  
7

дробная часть  
507500000

i=8 j=2

1 1 9

целая часть  
0

дробная часть  
604080000

i=8 j=3

1 1 9

целая часть  
0

дробная часть  
892350000

i=9 j=1

1 1 9

целая часть  
6

дробная часть

218399999

i=9 j=2

1 1 9

целая часть

0

дробная часть

917380000

i=9 j=3

1 1 9

целая часть

0

дробная часть

473130000

i=10 j=1

1 1 9

целая часть

3

дробная часть

898899999

i=10 j=2

1 1 9

целая часть

0

дробная часть

582330000

i=10 j=3

1 1 9

целая часть

0

дробная часть

235790000

i=11 j=1

1 1 9

целая часть

6

дробная часть

616909999

i=11 j=2

1 1 9

целая часть

0

дробная часть

173760000

i=11 j=3

1 1 9

целая часть

0

дробная часть

920470000

i=12 j=1

1 1 9

целая часть

6

дробная часть

400579999

i=12 j=2

1 1 9

целая часть

0

дробная часть

959020000

i=12 j=3

1 1 9

целая часть

0

дробная часть

469880000

i=13 j=1

1 1 9

целая часть

6

дробная часть

483970000

i=13 j=2

1 1 9

целая часть  
0

дробная часть  
188040000

i=13 j=3

1 1 9

целая часть  
0

дробная часть  
899350000

i=14 j=1

1 1 9

целая часть  
4

дробная часть  
000650000

i=14 j=2

1 1 9

целая часть  
0

дробная часть  
703770000

i=14 j=3

1 1 9

целая часть  
0

дробная часть  
128420000

i=15 j=1

1 1 9

целая часть  
4

дробная часть  
391470000

i=15 j=2

1 1 9

целая часть  
0

дробная часть  
436620000

i=15 j=3

1 1 9

целая часть  
0

дробная часть  
463940000

i=16 j=1

1 1 9

целая часть  
8

дробная часть  
912029999

i=16 j=2

1 1 9

целая часть  
0

дробная часть  
854060000

i=16 j=3

1 1 9

целая часть  
0

дробная часть  
948920000

i=17 j=1

1 1 9

целая часть  
3

дробная часть  
765760000

i=17 j=2

1 1 9

целая часть  
0

дробная часть  
007000000

i=17 j=3

1 1 9

целая часть  
0

дробная часть  
501730000

i=18 j=1

1 1 9

целая часть  
5

дробная часть  
166210000

i=18 j=2

1 1 9

целая часть  
0

дробная часть  
947420000

i=18 j=3

1 1 9

целая часть  
0

дробная часть  
168960000

i=19 j=1

1 1 9

целая часть  
4

дробная часть  
245930000

i=19 j=2

1 1 9

целая часть  
0

дробная часть  
486950000

i=19 j=3

1 1 9

целая часть  
0

дробная часть  
401070000

i=20 j=1

1 1 9

целая часть  
3

дробная часть  
946190000

i=20 j=2

1 1 9

целая часть  
0

дробная часть  
023870000

i=20 j=3

1 1 9

целая часть  
0

дробная часть  
533560000

i=21 j=1

1 1 9

целая часть  
5

дробная часть  
208109999

i=21 j=2

1 1 9

целая часть  
0

дробная часть  
986510000

i=21 j=3

1 1 9

целая часть  
0

дробная часть  
117040000

i=22 j=1

1 1 9

целая часть  
8

дробная часть  
324930000

i=22 j=2

1 1 9

целая часть  
0

дробная часть  
823630000

i=22 j=3

1 1 9

целая часть  
0

дробная часть  
888400000

i=23 j=1

1 1 9

целая часть  
6

дробная часть  
406600000

i=23 j=2

1 1 9

целая часть  
0

дробная часть  
917740000

i=23 j=3

1 1 9

целая часть  
0

дробная часть  
510810000

i=24 j=1

1 1 9

целая часть  
4

дробная часть  
734400000

i=24 j=2

1 1 9

целая часть  
0

дробная часть  
805220000

i=24 j=3

1 1 9

целая часть  
0

дробная часть  
234050000

i=25 j=1

1 1 9

целая часть  
5

дробная часть  
985560000

i=25 j=2

1 1 9

целая часть  
0

дробная часть  
157310000

i=25 j=3

1 1 9

целая часть  
0

дробная часть  
837430000

i=26 j=1

1 1 9

целая часть  
3

дробная часть  
622739999

i=26 j=2

1 1 9

целая часть  
0

дробная часть  
608830000

i=26 j=3

1 1 9

целая часть  
0

дробная часть  
116070000

i=27 j=1

1 1 9

целая часть  
3

дробная часть  
023629999

i=27 j=2

1 1 9

целая часть  
0

дробная часть  
216960000

i=27 j=3

1 1 9

целая часть  
0

дробная часть  
257110000

i=28 j=1

1 1 9

целая часть

3

дробная часть  
891960000

i=28 j=2

1 1 9

целая часть  
0

дробная часть  
590020000

i=28 j=3

1 1 9

целая часть  
0

дробная часть  
226140000

i=29 j=1

1 1 9

целая часть  
2

дробная часть  
720419999

i=29 j=2

1 1 9

целая часть  
0

дробная часть  
046680000

i=29 j=3

1 1 9

целая часть  
0

дробная часть  
244800000

i=30 j=1

1 1 9

целая часть  
2

дробная часть

255259999

i=30 j=2

1 1 9

целая часть  
0

дробная часть  
048710000

i=30 j=3

1 1 9

целая часть  
0

дробная часть  
089020000

i=31 j=1

1 1 9

целая часть  
5

дробная часть  
455230000

i=31 j=2

1 1 9

целая часть  
0

дробная часть  
095760000

i=31 j=3

1 1 9

целая часть  
0

дробная часть  
773380000

i=32 j=1

1 1 9

целая часть  
6

дробная часть  
926110000

i=32 j=2

1 1 9

целая часть

0

дробная часть

778490000

i=32 j=3

1 1 9

целая часть

0

дробная часть

710510000

i=33 j=1

1 1 9

целая часть

2

дробная часть

741870000

i=33 j=2

1 1 9

целая часть

0

дробная часть

256630000

i=33 j=3

1 1 9

целая часть

0

дробная часть

145160000

i=34 j=1

1 1 9

целая часть

3

дробная часть

325460000

i=34 j=2

1 1 9

целая часть  
0

дробная часть  
561300000

i=34 j=3

1 1 9

целая часть  
0

дробная часть  
061370000

i=35 j=1

1 1 9

целая часть  
3

дробная часть  
571540000

i=35 j=2

1 1 9

целая часть  
0

дробная часть  
316530000

i=35 j=3

1 1 9

целая часть  
0

дробная часть  
346850000

i=36 j=1

1 1 9

целая часть  
3

дробная часть  
106549999

i=36 j=2

1 1 9

целая часть  
0

дробная часть  
232290000

i=36 j=3

1 1 9

целая часть  
0

дробная часть  
272110000

i=37 j=1

1 1 9

целая часть  
6

дробная часть  
670060000

i=37 j=2

1 1 9

целая часть  
0

дробная часть  
541990000

i=37 j=3

1 1 9

целая часть  
0

дробная часть  
803020000

i=38 j=1

1 1 9

целая часть  
6

дробная часть  
048919999

i=38 j=2

1 1 9

целая часть  
0

дробная часть  
940150000

i=38 j=3

1 1 9

целая часть  
0

дробная часть  
413740000

i=39 j=1

1 1 9

целая часть  
3

дробная часть  
314810000

i=39 j=2

1 1 9

целая часть  
0

дробная часть  
021090000

i=39 j=3

1 1 9

целая часть  
0

дробная часть  
402910000

i=40 j=1

1 1 9

целая часть  
5

дробная часть  
450940000

i=40 j=2

1 1 9

целая часть  
0

дробная часть  
227620000

i=40 j=3

1 1 9

целая часть  
0

дробная часть  
739520000

i=41 j=1

1 1 9

целая часть  
5

дробная часть  
412020000

i=41 j=2

1 1 9

целая часть  
0

дробная часть  
388570000

i=41 j=3

1 1 9

целая часть  
0

дробная часть  
675760000

i=42 j=1

1 1 9

целая часть  
3

дробная часть  
910690000

i=42 j=2

1 1 9

целая часть  
0

дробная часть  
557400000

i=42 j=3

1 1 9

целая часть  
0

дробная часть  
262550000

i=43 j=1

1 1 9

целая часть  
8

дробная часть  
099619999

i=43 j=2

1 1 9

целая часть  
0

дробная часть  
558750000

i=43 j=3

1 1 9

целая часть  
0

дробная часть  
989520000

i=44 j=1

1 1 9

целая часть  
6

дробная часть  
536299999

i=44 j=2

1 1 9

целая часть  
0

дробная часть  
908340000

i=44 j=3

1 1 9

целая часть  
0

дробная часть  
544380000

i=45 j=1

1 1 9

целая часть  
4

дробная часть  
663669999

i=45 j=2

1 1 9

целая часть  
0

дробная часть  
091240000

i=45 j=3

1 1 9

целая часть  
0

дробная часть  
648010000

i=46 j=1

1 1 9

целая часть  
4

дробная часть  
187739999

i=46 j=2

1 1 9

целая часть  
0

дробная часть  
066930000

i=46 j=3

1 1 9

целая часть  
0

дробная часть  
569470000

i=47 j=1

1 1 9

целая часть

7

дробная часть

404280000

i=47 j=2

1 1 9

целая часть

0

дробная часть

814470000

i=47 j=3

1 1 9

целая часть

0

дробная часть

761570000

i=48 j=1

1 1 9

целая часть

4

дробная часть

526609999

i=48 j=2

1 1 9

целая часть

0

дробная часть

239130000

i=48 j=3

1 1 9

целая часть

0

дробная часть

580710000

i=49 j=1

1 1 9

целая часть

5

дробная часть  
845060000

i=49 j=2

1 1 9

целая часть  
0

дробная часть  
332070000

i=49 j=3

1 1 9

целая часть  
0

дробная часть  
766030000

i=50 j=1

1 1 9

целая часть  
6

дробная часть  
537939999

i=50 j=2

1 1 9

целая часть  
0

дробная часть  
607500000

i=50 j=3

1 1 9

целая часть  
0

дробная часть  
750740000

степень полинома=1

зависимый параметр-1

(p 1)  
номера аргументов  
2 3

оптимальная степень=1

Модель для зависимого параметра

аргумент-2

(p 2)

номер аргумента-2 степень=1

знак=1

целая часть коэффициента

3

дробная часть коэффициента

02957905235853855252

аргумент-3

(p 3)

номер аргумента-3 степень=1

знак=1

целая часть коэффициента

5

дробная часть коэффициента

05199115315277216821

свободный член модели

знак свободного члена=1

целая часть свободного члена

1

дробная часть свободного члена

14626064145681688366

средняя абсолютная ошибка

целая часть ошибки

0

дробная часть ошибки

17342845830972969901

номер аргумента=2

(p 2)

целая часть вклада

0

дробная часть вклада

36555678683089752035

номер аргумента=3

(p 3)

целая часть вклада

0

дробная часть вклада

63444321316910247964

**Протокольный файл \_outfile**

Start date: 2008.08.16 16:19:31 YEKST

pr i=2

pr i=3

End date: 2008.08.16 16:19:33 YEKST