

Министерство науки и высшего образования Российской Федерации

Университетский колледж
федерального государственного бюджетного образовательного учреждения
высшего образования
«Оренбургский государственный университет»

Отделение информационных технологий

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Методические указания

Составители: С.В. Ханжин, Л.А. Грекова

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательной программе среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах

Оренбург
2021

УДК 681.3.06 (075.32)
ББК 22.18 я723
О 75

Рецензент – доцент кафедры прикладной информатики в экономике и управлении И.А. Цыганова

О 75 **Основы программирования:** методические указания / составители:
Л.А. Грекова, С.В. Ханжин; Оренбургский гос. ун-т. – Оренбург: ОГУ,
2021. – 43 с.

Методические указания предназначены для выполнения лабораторных работ по дисциплине «Основы программирования» в Университетском колледже ОГУ для обучающихся второго курса специальности 09.02.03 Программирование в компьютерных системах.

Методические указания составлены с учетом Федерального государственного образовательного стандарта среднего профессионального образования.

УДК 681.3.06(075.32)
ББК 22.18я723

© Грекова Л.А, Ханжин С.В.,
составление 2021
© ОГУ, 2021

Содержание

Введение	4
1 Лабораторная работа «Программирование с использованием одномерных массивов»	6
2 Лабораторная работа «Программирование с использованием динамических двумерных массивов»	15
3 Лабораторная работа «Программирование с использованием строк»	23
Список использованных источников	34
Приложение А.....	35
Приложение Б	36
Приложение В.....	38
Приложение Г	41

Введение

В программировании чаще всего приходится работать с очень большими объемами данных. Для выполнения каких-либо прикладных задач требуется обработка информации, представленной в различных видах. В связи с этим рациональным будет представление информации в виде различных массивов и строк. Такое представление обуславливается тем, что для массивов и строк разработано большое количество различных алгоритмов для их обработки.

Методические указания позволят на конкретных примерах изучить различные способы обработки массивов и строк.

Цель данных методических указаний состоит в возможности показать некоторые способы решения различных задач с использованием одномерных, двумерных массивов и строк в языке программирования высокого уровня C++.

Решаемые прикладные задачи сгруппированы в три лабораторные работы по соответствующим темам: Программирование с использованием одномерных массивов, Программирование с использованием динамических двумерных массивов, Программирование с использованием строк. На каждое лабораторное занятие отводится по 4 академических часа.

Структура задания на каждое лабораторное занятие следующая:

1. Цель работы
2. Ход работы
3. Содержание отчёта
4. Методические указания к лабораторной работе
5. Варианты индивидуальных заданий.

Методы обучения: практический, наглядный, словесный, формирование умений, проблемный.

Оборудование, необходимое для проведения лабораторных работ:

- мультимедийный комплекс;

- персональные компьютеры.

Методическое обеспечение лабораторных занятий: презентация; среда программирования DevC++; видеоролики; методические указания; дидактический материал.

Для оценивания результатов лабораторного занятия предлагается Таблица критериев оценивания (Приложение А).

Для примера правильного оформления отчета приведено оформление отчета по нулевому варианту лабораторной работы «Программирование с использованием одномерных массивов» (Приложение Б).

1 Лабораторная работа «Программирование с использованием одномерных массивов»

Цель работы. Научиться правильно описывать одномерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научиться решать задачи на использование массивов.

1.1 Ход работы

- 1) Изучить теоретический материал по теме лабораторной работы (лекции, учебник).
- 2) Составить программы решения задач своего варианта.
- 3) Составить отчёт.
- 4) Защитить работу.

1.2 Содержание отчёта

Отчет по лабораторной работе должен содержать:

- 1) Тема работы.
- 2) Цель работы.
- 3) Ход работы.
- 4) Постановка задачи.
- 5) Программа.
- 6) Вывод.

1.3 Методические указания к лабораторной работе «Программирование с использованием одномерных массивов»

Массив – это упорядоченная совокупность данных, имеющая одно имя и один тип.

Упорядоченность массива заключается в наличии индекса- порядкового номера массива.

Чтобы объявить массив в программы следует использовать квадратные скобки [], что являться частью синтаксиса, а не признаком необязательности конструкции. Число, записанное в квадратных скобках –размер массива.

Размерность массива –число индексов массива (одномерные, двумерные и т.д.)

Примером массив может быть книга: книга имеет одно название, что я является именем массива, все страницы книги пронумерованы по порядку, что показывает наличие индекса массива, и все страницы — книги- это однотипные элементы.

Особенностью индексов массива языка C++ является то, что нумерация начинается с нулевого элемента.

Объявление массива имеет следующий синтаксис:

<спецификация типа><имя> [<константное выражение>];

<спецификация типа><имя>[];

intsum_ozenok[13]; // Одномерный массив из 13 целых чисел

doubleproisz[20][15]; // Двумерный массив вещественных чисел из 20 строк и 15 столбцов.

Для обработки элементов массива целесообразно использовать цикл с параметром, так как параметр цикла изменяется так же, как порядковый номер(индекс) массива.

Ввод элементов массива, состоящего из m элементов:

```
intm;
```

```
cin>> m;
```

```
for (inti = 0; i<m; i++)
```

```
cin>>masiv_1[i];
```

Пример программы с использованием массива

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     setlocale(LC_ALL, "rus");
6
7     const int n = 10;
8     int arr[n];
9     int sum = 0;
10    int count = 0;
11    int average;
12
13    for (int i = 0; i < n; i++) {
14        cout << "[" << i << "]" << ": ";
15        cin >> arr[i];
16    }
17
18    cout << "\nЧетные: ";
19    for (int i = 0; i < n; i += 2) {
20        cout << arr[i] << " ";
21        sum += arr[i];
22        count++;
23    }
24
25    cout << "\nНечетные: ";
26    for (int i = 1; i < n; i += 2) {
27        cout << arr[i] << " ";
28    } cout << endl;
29
30    average = sum / count;
31    cout << "\nСр.Арифмитическое = " << average << endl;
32
33    return 0;
34 }
```

Рисунок 1- Программа с использованием одномерных массивов

Результат работы программы:

```
[0]: 1
[1]: 2
[2]: 3
[3]: 4
[4]: 5
[5]: 6
[6]: 7
[7]: 8
[8]: 9
[9]: 10

Четные: 1 3 5 7 9
Нечетные: 2 4 6 8 10

Ср.Арифмитическое = 5
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2 – Результат программы

1.4 Варианты индивидуальных заданий

Индивидуальное задание № 1

Составить программу по заданию своего варианта.

1) В массиве, состоящем из 10 целых чисел подсчитать сумму элементов больше 0, произведение элементов меньше нуля и количество элементов равных нулю. Вывести элементы массива в обратном порядке.

2) Массив `Max_element` заполнить случайными числами в диапазоне от -10 до 100 вычислить среднее арифметическое элементов, которые больше нуля.

3) Дан массив `Massiv_index` размера `N`. Вывести элементы с четными индексами, а элементы с нечетными индексами умножить на 2. Отсортировать элементы массива методом «пузырька»

4) В массиве `Proizv_mas` размером 100 есть хотя бы один нулевой элемент. Вычислить произведение элементов, стоящих до первого нулевого элемента и сумму элементов после нулевого элемента.

5) В массив `Max_element` размером `N` вычислить сумму элементов стоящих на четных местах. Определить минимальный элемент массива и его индекс, результат вывести на экран.

6) Дан массив `Masiv_1`, состоящий из 100 элементов. Сформировать массив `Masivs_2`, каждый элемент которого вычисляется по формуле $mas_i = e^{a[i]-i}$. Отсортировать массив `Masivs_2[i]` в порядке возрастания.

7) Используя генератор случайных чисел заполнить массив `kol_mas` случайными числами в диапазоне от -10 до 10. Вывести на экран положительные элементы массива, элементы массива меньше нуля умножить на 10 и вывести на экран.

8) Дан массив Z_mas_n , заполненный элементами целого типа. Сформировать массив Y_mas_n , каждый элемент которого заполняется по формуле $Y_mas_n = Z_mas_n / \ln Z_n$. Организовать вывод обоих массивов. Сравнить максимальные и минимальные элементы массивов Z_mas_n и Y_mas_n

9) Массив kol_mas , заполнить случайными числами в диапазоне от -20 до 20. Используя алгоритм сортировки, упорядочить элементы по возрастанию. Все положительные элементы умножить на 2 и вывести на экран.

10) Сформировать массив PR_n , каждый элемент которого определяется по формуле $PR_k = K * \cos(A_k)$. Используя алгоритм сортировки расположить все элементы массива в порядке возрастания.

11) В целочисленном массиве размером m заменить положительные элементы на 0, отрицательные элементы на 1, вывести полученный массив на экран и подсчитать кол – во 0 и 1.

12) Дан массив $vass_1$ размера N . Вычислить сумму элементов массива. Если сумма больше 0, то каждый элемент массива возвести в третью степень.

13) Дан массив Pol_k размера N . Сформировать массив Zen_k , каждый элемент которого определяется по формуле $Zen_k = P_k$. Найти максимальный элемент массива Zen_k и заменить элементы, стоящие на нечетных позициях в массиве Pol_k на минимальный элемент массива Zen_k .

14) Даны два массива Pol_k и Otr_k размерам. Сформировать третий массив Sen_k , элементы которого получаются путем сложения соответствующих элементов массива Pol_k и Otr_k .

15) Массив Sum_mas заполнить случайными числами в диапазоне от -10 до 50. Каждый элемент массива, который больше нуля возвести в квадрат. Найти сумму отрицательных элементов массива Sum_mas .

16) Дан массив, состоящий из элементов вещественного типа. Вывести элементы массива в порядке убывания. После сортировки заменить отрицательные элементы на 0, положительные на 1. Найти произведение отрицательных элементов.

17) Массив `Kol_el` заполнить случайными числами. Найти среднее арифметическое значение всех элементов массива и вывести на экран, те элементы массива, значения которых больше среднего арифметического значения.

18) В массиве `Osenki` записаны все оценки за экзамен по программированию. Подсчитать сколько студентов сдали экзамен на “5”, на “4”, на “3”, и вывести значения на экран.

19)) Дан массив `Masn`, заполненный случайными элементами в диапазоне от 0 до 100. Вывести на экран максимальный элемент массива и его индекс. Упорядочить элементы массива по убыванию, используя «пузырьковый» метод сортировки массива.

20) В массиве в `Dev_1`, заполненный случайными числами подсчитать сумму положительных элементов стоящих на нечетных местах. Вычислить произведение четных элементов массива.

21) Массив размера `m` заполнить, используя оператор потокового ввода. Вычислить среднее арифметическое и количество отрицательных элементов массива. Заменить в массиве все нули на единицы.

22) В целочисленном массиве `kol_mas`, подсчитать количество и произведение элементов до первого отрицательного.

23) В массиве, состоящем из 100 целых чисел подсчитать сумму элементов больше 10, произведение элементов меньше нуля и количество элементов равных нулю. Вывести элементы массива в порядке убывания.

24) Дан массив размером N . Вычислить сумму элементов стоящих на нечетных местах. Отсортировать по убыванию.

25) Дан целочисленный массив Cum_i размером N . Сформировать массив Vum_i элементы которого вычисляются по формуле $Vum_i=(i+1)*2$. Для массива Vum_i подсчитать количество и сумму элементов до первого отрицательного.

Индивидуальное задание № 2

Составить программу по заданию своего варианта.

1. Дан массив Mas_chislo , состоящий из 100 вещественных чисел. Вывести в обратном порядке по 8 чисел в строке.

2. Дан массив Mas_A , состоящий из 100 целых чисел. Вывести на экран сначала отрицательные из них, а затем положительные.

3. В массиве Z_mas размера N заменить все положительные элементы максимальным значением массива. Организовать ввод элементов массива потоковым вводом с клавиатуры.

4. Дан символьный массив, размера N . Подсчитайте максимальное количество подряд идущих пробелов.

5. Пусть дано натуральное число N - число элементов массива. Необходимо всю последовательность элементов умножить на 2 и вывести на экран наибольший элемент последовательности.

6. В целочисленном массиве M_1 , организовать заполнение массива случайными элементами. Вывести на экран сумму положительных, произведение отрицательных и число нулевых членов последовательности .

7. Пусть дана последовательность из 100 различных целых чисел. Найдите средние арифметическое чисел этой последовательности, расположенных между максимальным и минимальным числами.

8. Дан массив из N элементов. Каждый положительный элемент замените полусуммой максимального и минимального элемента массива .

9. В вещественном массиве MAX_MIN заменить все отрицательные элементы суммой максимального и минимального элементов.
10. Создать массив K_mas из 20 элементов. Первые 10 элементов необходимо сложить, вторые 10 перемножить между собой. Вычислить среднеарифметическое значение суммы и произведения элементов.
11. Создать массив, содержащий больше 25 элементов. Поменять местами первый и последний элемент массива.
12. В массиве Zet максимальным числом ε которого является 10 элементов вещественного типа необходимо выполнить сортировку от большего числа к наименьшему.
13. Дан массив Vas[100], необходимо заполнить случайными числами в диапазоне от 25 до 50. Найти среднеарифметическое всех отрицательных чисел данного массива.
14. Дан массив размера N. Необходимо заменить каждое число, стоящее на нечетной позиции на наибольший элемент в массиве.
15. Даны числа от 0 до 9 необходимо задать каждой цифре буквенное соответствие.
16. В группе 25 студентов. Необходимо создать целочисленный массив и каждой цифре данного массива присвоить буквенное обозначение. Поменять местами 12 и 14 элемент.
17. В массиве из 9 элементов необходимо вывести четные и нечетные элементы. Отсортировать все элементы в порядке убывания.
18. В массиве Sred найти среднеарифметическое значение всех положительных элементов. Элементу с индексом 0 присвоить значение среднеарифметическое значение.
19. Дан целочисленный массив DEV_i размером N. Сформировать новый массив элементы которого вычисляются по формуле $(i*5)*10$.
20. Дан массив Pricv[] и константное значение переменной, необходимо увеличить каждый элемент массива Pricv[] на заданное константное значение.

1.5 Вопросы к защите лабораторной работы «Программирование с использованием одномерных массивов»

- 1) Что такое размер и размерность массива?
- 2) Какие действия можно производить над массивом целиком и при каких условиях?
- 3) В чем особенность обращения к элементу массива?
- 4) Какие операции можно производить над элементами массива?
- 5) Перечислите способы ввода элементов массива?
- 6) Почему для обработки элементов массива целесообразно использовать цикл с параметром?
- 7) В чем заключается «пузырьковый» алгоритм сортировки массива?

2 Лабораторная работа «Программирование с использованием динамических двумерных массивов»

Цель работы. Научиться правильно описывать двумерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научиться решать задачи на использование массивов.

2.1 Ход работы

- 1) Изучить теоретический материал по теме лабораторной работы (лекции, учебник).
- 2) Составить блок-схемы и программы решения задач своего варианта.
- 3) Составить отчёт.
- 4) Защитить работу.

2.2 Содержание отчёта

Отчет по лабораторной работе должен содержать:

- 1) Тема работы.
- 2) Цель работы.
- 3) Ход работы.
- 4) Постановка задачи.
- 5) Блок-схема алгоритма.
- 6) Программа.
- 7) Вывод.

2.3 Методические указания к лабораторной работе «Программирование с использованием динамических двумерных массивов»

Массив – это упорядоченная совокупность данных, имеющая одно имя и один тип.

Упорядоченность массива заключается в наличии индекса- порядкового номера массива.

Чтобы объявить массив в программы следует использовать квадратные скобки [], что являться частью синтаксиса, а не признаком необязательности конструкции. Число, записанное в квадратных скобках –размер массива.

Размерность массива –число индексов массива (одномерные, двумерные и т.д.)

Примером двумерного массив может быть шахматная, так как доска имеет одно имя, на поле перемещаются однотипные элементы, каждое поле имеет двойной адрес(номер строки и номер столба на пересечении которого он находится)

Многомерным массивом называют массив массивов.

```
Int MAS_ARREY[5][7];
```

Поскольку у нас есть 2 индекса, то это **двумерный массив**.

В двумерном массиве первый (левый) индекс означает количество строк, а второй (правый) означает количество столбцов.

Массив выше можно представить следующим образом:

```
[0][0] [0][1] [0][2] [0][3] // строка №0
```

```
[1][0] [1][1] [1][2] [1][3] // строка №1
```

Чтобы получить доступ к элементам двумерного массива — просто используйте два индекса:

```
Array_mas[2][3] = 17;
```

Для инициализации двумерного массива используются вложенные фигурные скобки, где построчно указываются элементы массива:

```
intarray[3][5] =
```

```

{
{ 1, 2, 3, 4, 5 },      // строка №0
{ 6, 7, 8, 9, 10 },   // строка №1
{ 11, 12, 13, 14, 15 } // строка №2
};

```

C++ заменяет отсутствующие инициализаторы значением 0:

```

intarray[3][5] =
{
{ 2, 4 },              // строка №0 = 2, 4, 0, 0, 0
{ 1, 3, 7 },          // строка №1 = 1, 3, 7, 0, 0
{ 8, 9, 11, 12 }     // строка №2 = 8, 9, 11, 12, 0
};

```

В двумерном массиве со списком инициализаторов можно не указывать только левый индекс (длину массива):

```

intarray[][2] =
{
{ 1, 2, 3, 4, 5 },
{ 6, 7, 8, 9, 10 },
};

```

Для обработки элементов двумерного массива требуется двойной цикл, так первый цикл отвечает за обработку элементов по строкам, а второй по столбцам. Поскольку доступ к двумерным массивам обычно выполняется по строкам, то левый индекс используется в качестве внешнего цикла:

```

for (int red = 0; row < numred; ++red) //обработка по строкам
    for (int der = 0; col < numder; ++der) //обработка каждого элемента в строке
        cout << array[red][der];

```

Пример.

```
ConsoleApplication1 (Глобальная)
5 int main() {
6     setlocale(LC_ALL, "rus");
7     srand((time(NULL)));
8     int** mat;
9     int M, sum = 0, sum1 = 0;
10    cout << "Введите M: ";
11    cin >> M;
12    mat = new int* [M];
13    cout << "Матрица: " << endl;
14    for (int i = 0; i < M; i++)
15    {
16        mat[i] = new int[M];
17        for (int j = 0; j < M; j++)
18        {
19            mat[i][j] = rand() % 9;
20            cout << mat[i][j] << " ";
21            if (i == j)
22            {
23                sum += mat[i][j];
24            }
25            if (i + j == M - 1)
26            {
27                sum1 += mat[i][j];
28            }
29        }
30        cout << endl;
31    }
32    cout << "Сумма эл-тов гл. диагонали: " << sum << endl;
33    cout << "Сумма эл-тов поб. диагонали: " << sum1 << endl;
34    if (sum > sum1)
35    {
36        cout << "Сумма эл-тов гл. диагонали больше суммы эл-тов поб. диагонали.";
37    }
38    else
39    {
40        cout << "Сумма эл-тов поб. диагонали больше суммы эл-тов гл. диагонали.";
41    }
42    cout << endl;
43    return 0;
44 }
```

Рисунок 3 –Пример программы с использованием двумерных массивов

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      setlocale(LC_ALL, "rus");
6
7      int N = 23;
8      int M = 17;
9      float** arr;
10     arr = new float* [N];
11     for (int i = 0; i < N; i++)
12     {
13         arr[i] = new float[M];
14     }
15     cout << "Матрица: \n";
16     for (int i = 0; i < N; i++)
17     {
18         for (int j = 0; j < M; j++)
19         {
20             arr[i][j] = (double)1/(i + j);
21             cout.precision(2);
22             cout << fixed << arr[i][j] << "\t";
23         }
24         cout << endl;
25     }
26
27     return 0;
28 }

```

Рисунок 3 – Пример программы формирования вещественной матрицы A размером 23 x 17, для которой: $a_{ij} = \frac{1}{i+j}$.

2.4 Варианты индивидуальных заданий

Индивидуальное задание № 1

Составить блок-схему и программу по заданию своего варианта.

1) Сформировать квадратную матрицу Matrisa размером N, заполнить положительными значениями целого типа. Все элементы матрицы, которые лежат на главной диагонали заменить нулевыми значениями. Вывести на экран количество ненулевых элементов матрицы.

2) В матрице mas_sum размером M*N рассчитать и вывести на экран сумму и число положительных элементов, находящихся над главной диагональю.

3) В квадратной матрице размера N произвести сравнение суммы элементов главной и побочной диагонали.

4) В квадратной матрице `Massiv` размера M произвести замену отрицательных элементов матрицы на нулевые значения и вычислить произведение положительных элементов.

5) Матрицу размера $M*N$ заполнить случайными числами в диапазоне от -10 до 10 и произвести сравнение суммы элементов четных строк с суммой элементов нечетных столбцов.

6) В матрице произвольного размера произвести упорядочивание элементов каждой строки по возрастанию.

7) Дана матрица `mas_sred` размера $M*N$, необходимо вычислить среднее арифметическое значение в каждой строке и максимальное значение в каждом столбце.

8) В матрице произвольного размера произвести упорядочивание элементов по возрастанию в каждой строке.

9) В целочисленной матрице `Mas_tsel` найти максимальный и минимальный элемент, и поменять их местами.

10) Определить индексы наибольшего и наименьшего элементов квадратной матрицы, заполненной случайными числами.

11) Сформировать матрицу в виде таблицы умножения.

12) В целочисленной матрице размера $M*N$ необходимо в каждой строке вывести на экран элемент с наименьшим значением, затем среди этих чисел вывести наибольшее значение.

13) В двумерном массиве `Mas_chet[3][5]` необходимо вывести на экран сумму всех нечетных элементов и сумму всех четных элементов.

14) Создайте матрицу `Matr_t[7][7]`, заполненную случайными значениями в диапазоне от 0 до 100. Вывести на экран индексы максимального и минимального элементов.

15) В квадратной матрице размером M заменить все отрицательные элементы числом A , а все положительные элементы числом B .

16) В двумерном массиве `mas_ct[6][5]` поменять местами элементы первого и последнего столбца

Индивидуальное задание № 2

Составить программу по заданию своего варианта.

1) Даны целые числа x_1, x_2, x_3 . Сформировать массив z размером 3×3 , для которой $z_{ij} = x_i - 3x_j$.

2) Массив D размер 3×5 заполнить с клавиатуры целыми числами. Необходимо сложить все элементы, стоящие на главной диагонали, а элементы, стоящие на побочной диагонали умножить на сумму элементов первого столбца.

3) В матрице `Maz_st` 7×5 умножить каждый элемент матрицы минимальный элемент, если он больше суммы элементов первой строки.

4) Сформировать массив `sl_mas[6][5]` из случайных чисел произвольного диапазона. Вывести на экран элементы массива, которые больше нуля и кратны L , введённому с клавиатуры.

5) В матрице размером 5×5 каждый элемент главной диагонали умножить на произведение элементов последней строки

6) Дана квадратная целочисленная матрица размером 5×5 , заполненная случайными значениями произвольного диапазона. Если минимальный элемент третьей строки не равен 0, то разделить на него все остальные элементы.

7) Сформировать двумерный массив 5×7 (5 строк, 7 столбцов) элементами которого являются целые случайные числа в пределах от 1 до 45. Найти минимальный элемент и номер минимального элемента.

8) Сформировать таблицу Пифагора 10×10 (таблица умножения) и вывести ее на экран.

9) Заполнить матрицу 5×5 любым способом. Если максимальный элемент третьей строки не равен 0, то разделить на него все остальные элементы.

10) В матрице `Strok_mas` 6×6 поменять местами 1 и 2 строку, 3 и 4 строку, 5 и 6 строку. Воспользоваться вспомогательным одномерным массивом.

11) В квадратной матрице Mas_KV [7][7] найти наибольший элемент главной диагонали, наименьший элемент побочной, произвести сравнение этих значений и вывести результат на экран.

12) В целочисленной матрице per_st 4*4 найти максимальный элемент первой строки и посчитать, сколько элементов массива больше этого значения.

13) Задан двумерный массив Bas [7][7] заполненный случайными числами из диапазона [-10,10]. Найти и вывести на экран те элементы массива, которые больше заданного числа k.

14) В целочисленной матрице порядка 4x6 поменять местами элементы первого и последнего столбца.

15) Задан двумерный массив 4x5, умножить каждый элемент матрицы на максимальный элемент, если он больше суммы элементов первой строки.

16) Массив, состоящий из трех строк и пяти столбцов заполнить случайными числами в диапазоне -100 до 100. Найти сумму элементов массива имеющих нечётное значение.

2.5 Вопросы к защите лабораторной работы «Программирование с использованием двумерных массивов»

- 1) Перечислить способы заполнения двумерного массива?
- 2) Привести примеры двумерного массива из любой предметной области?
- 3) Как определить местоположение элемента в двумерном массиве?
- 4) Какие действия можно производить над массивом целиком и при каких условиях?
- 5) Как указать диапазон значений для случайного заполнения элементов в двумерном массиве?
- 6) Как находить элементы, стоящие на главной и побочной диагонали?
- 7) Какими правилами нужно руководствоваться, чтобы задавать имя массиву?

3 Лабораторная работа «Программирование с использованием строк»

Цель работы. Получение навыков в организации ввода/вывода значений строковых типов данных. Получение практических навыков программирования задач с использованием строк.

3.1 Ход работы

- 1) Изучить теоретический материал по теме лабораторной работы (лекции, учебник).
- 2) Составить программу решения задач своего варианта.
- 3) Составить отчёт.
- 4) Защитить работу.

3.2 Содержание отчёта

Отчет по лабораторной работе должен содержать:

- 1) Тема работы.
- 2) Цель работы.
- 3) Ход работы.
- 4) Постановка задачи.
- 5) Программа.
- 6) Вывод.

3.3 Методические указания к лабораторной работе «Программирование с использованием строк»

Достаточно часто в C++ используются строки для представления текста (предложений, адресов, имён, слов). Строковые константы помещаются в двойные кавычки.

В C++ существует стандартный тип **string**, но не как часть самого языка, а только как часть стандартной библиотеки.

К любому отдельному символу строки можно обратиться по его индексу. Например, `Str[0]` - первый символ строки `Str`.

В случае, если необходимо узнать длину строки возможно использовать метод `size()`. Например, последний символ строки `Str` это `Str[Str.size() - 1]`.

Для того, чтобы получить возможность использовать строки в C++, в начале программы нужно подключить заголовочный файл **string**.

```
#include <string>
// ...
String name;
// ...
```

Как и для обычных переменных, существует возможность инициализации переменных типа `string` и присваивания им необходимых значений:

```
String surname("Petrov"); // инициализируем переменную surname строковой константой "Petrov"
```

```
    surname = "Ivanov"; // присваиваем переменной surname строковую константой "Ivanov"
```

Строки также могут содержать числа:

```
stringmyID("57"); // "57" в этом случае не целое число 57, а строка из двух символов!
```

Рассмотрим некоторые методы, которые позволяют выполнять различные действия со строками:

- 1) **size**

Метод **size()** возвращает длину строки.

2) **length**

Метод **length()** также возвращает длину строки.

3) **resize**

Str.resize(n) - Изменяет длину строки **Str**, новая длина строки **Str** становится равна **n**. При использовании этого метода строка **Str** может как уменьшаться, так и увеличиваться.

Если вызывать метод в виде **Str.resize(n, a)**, где **a** - символ, то увеличение длины строки будет происходить благодаря добавлению символов **a**.

4) **clear**

Str.clear() – данный метод очищает строку и строка становится пустой.

5) **empty**

Str.empty() – метод **empty** возвращает значение **true**, если строка пуста и **false** - если она не является пустой.

6) **push_back**

Str.push_back(a) – этот метод добавляет в конец строки символ **a**, используется с параметром типа **char**.

7) **append**

Метод **append** добавляет в конец строки последовательность символов, какую-либо другую строку или фрагмент строки. Имеет несколько различных способов вызова.

Str.append(N, f) - добавляет в конец строки **Str** **N** одинаковых символов **f**. **N** - целочисленный тип, **f** – тип **char**.

Str.append(ST) - добавляет в конец строки **Str** строку **ST**. **ST** может быть объектом класса **string**.

Str.append(ST, index, N) - добавляет в конец строки **Str** строку **ST**, начиная с символа с индексом **index**, количеством **N**.

8) **erase**

Str.erase(index) – удаляет символы из строки **Str**, начиная с символа с индексом **index** до конца строки.

`Str.erase(index, N)` - удаляет из строки `Str` символы, начиная с символа с индексом `index`, количеством `N` или до конца строки, если `index + N > Str.size()`.

9) **insert**

Данный метод заключается в вставке в середину строки нескольких символов, другой строки или фрагмента другой строки.

`index` - позиция, начиная с которой вставляются символы.

`Str.insert(index, N, f)` - вставляет `N` одинаковых символов, равных `f`. `N` - целочисленный тип, `f` - `char`.

`Str.insert(index, ST)` - вставляет содержимое строки `ST`. `ST` может быть объектом класса `string` или `C`-строкой.

`Str.insert(index, ST, pos, N)` - вставляет символы строки `ST`, начиная с символа с индексом `pos` количеством `N`.

10) **substr**

`Str.substr(index)` – метод возвращает подстроку строки `Str`, начиная с символа с индексом `index` до конца строки.

`Str.substr(index, count)` – метод возвращает подстроку строки `Str`, начиная с символа с индексом `index` количеством `N` или до конца строки, в случае если `index + N > Str.size()`.

11) **replace**

Данный метод заменяет фрагмент исходной строки на несколько одинаковых символов, другую строку полностью или фрагмент другой строки.

Первые два параметра - числа: `index` и `N`. Из исходной строки удаляется `N` символов, начиная с символа с индексом `index`, и вместо них вставляются новые символы.

`Str.replace(index, N, p, f)` - вставляется `p` одинаковых символов `f`. `N` - целочисленный тип, `f` - `char`.

`Str.replace(index, N, ST)` - вставляется содержимое строки `ST`. `ST` может быть объектом класса `string` или `C`-строкой.

`Str.replace(index, N, ST, index2, N2)` - вставляются символы строки `ST`, начиная с символа с индексом `index` количеством `N`.

12) **find**

Метод ищет в исходной строке первое вхождение некоторой строки `str1`. Возвращается номер символа, начиная с которого далее идет подстрока `str1`. Если строка `str1` не обнаружена, то возвращается константа `string::nindex` (которая будет равна -1).

Если дано значение `index`, то поиск начинается с индекса `index`, то есть возвращаемое значение будет не меньше, чем `index`. Если значение `index` не указано, то оно принимается равным 0 – и в этом случае поиск происходит с начала строки.

`Str.find(str1, index = 0)` - ищет первое вхождение строки `str1`, начиная с позиции `index`. Если `index` не задан, поиск начинается с начала строки `Str`.

`Str.find(str1, index, N)` - ищет в исходной строке подстроку, состоящую из первых `N` символов строки `str1`. Значение `index` обязательно должно быть задано.

13) **rfind**

Данный метод обнаруживает последнее вхождение подстроки (случай "правого" поиска). Способы вызова метода аналогичны вызову метода `find`.

14) **find_first_of**

Метод обнаруживает в исходной строке первое появление любого из символов строки `str1`. Возвращается номер данного символа или значение `string::nindex`.

Если задано значение `index`, то поиск происходит с позиции `index`, то есть возвращаемое значение будет не меньше, чем `index`. Если значение `index` не указано, то автоматически оно равно 0 и поиск осуществляется с начала исходной строки.

`Str.find_first_of(str1, index = 0)` - найти первое вхождение любого символа строки `str1`, начиная с индекса `index`. Если значение `index` не дано, то начиная с первого символа строки `Str`.

15) **find_last_of**

Находит в исходной строке последнее вхождение любого из символов заданной строки `str1`. Способы вызова и возвращаемое значение такие же как у метода `find_first_of`.

16) **find_first_not_of**

Находит в исходной строке первое вхождение символа, отличного от символов входящих в строку `str1`. Способы вызова и возвращаемое такие же как у метода `find_first_of`.

17) `find_last_not_of`

Находит в исходной строке последнее вхождение символа, отличного от символов входящих в строку `str1`. Способы вызова и возвращаемое такие же как у метода `find_first_of`.

18) `c_str`

Метод возвращает указатель на область памяти, где хранятся символы исходной строки, возвращается значение типа `char*`. Возвращаемое значение возможно рассматривать как C-строку и применять в функциях, которые имеют возможность использовать на входе C-строку.

3.4 Варианты индивидуальных заданий

Индивидуальное задание № 1

Составить программу по заданию своего варианта

1. Создайте программу, считывающую число букв в слове, введенном с клавиатуры. Ввод осуществляется в цикле `while do`. Условие выхода из программы – ввод строки `'000'`.
2. Создайте программу, считающую число вхождений введенной с клавиатуры буквы во введенной строке.
3. Создайте программу, которая выводит заданную строку, каждый раз сокращая ее на один символ до тех пор, пока в выводимой строке не останется только один символ.
4. Составьте программу, определяющую количество слов в заданной строке. Слова отделяются одним пробелом.
5. Напишите программу, которая определяет, являются ли введенные с клавиатуры символы числом.

6. Необходимо ввести с клавиатуры два целых числа. Преобразуйте введенные числа в 2 строки, затем объедините эти числа в одну строку и выведите результат.

7. Составьте программу, которая удаляет из заданной строки все введенные с клавиатуры символы. Данный процесс организуйте в отдельную функцию DelChInStr,

8. Напишите программу, которая удаляет пробелы из заданной строки. Для удаления пробелов создайте отдельную процедуру NewStr и используйте в ней оператор Repeat и функцию Pos.

9. Составьте программу, которая сортирует символы исходной строки в порядке возрастания их номеров в ASCII-таблице. К примеру, если введено: 'АБВ', на выходе необходимо получить 'ВБА'.

10. Написать программу вычисляющую длину наиболее короткого слова в строке из трех слов, разделенных 1 пробелом.

11. Определить, какой из символов, первый или последний встречается в введенном слове чаще.

12. Дано существительное, которое оканчивается на "а". Выведите данное слово во всех возможных падежах.

13. Сколько букв "а" стоит в слове на нечетных местах?

14. Поменяйте во введенном слове все буквы "т" на пробелы.

15. Введены 2 слова. Замените местами буквы данных слов, которые занимают одни и те же позиции.

16. Введены фамилия, имя и отчество ученика, которые разделены пробелами. Выведите фамилию и инициалы.

17. Удалите k-ю букву слова (k вводится с клавиатуры).

18. Введен текст. Необходимо определить:

а) сколько слов в тексте?

б) найти самое короткое слово текста.

19. Дан текст, состоящий из слов, между которыми один или несколько пробелов. Выведите текст, состоящий из слов данного, но разделенных только одним пробелом.

20. Большое слово состоит из двух частей, одинаковых по длине и соединительной гласной. Выведите две части этого слова.

21. Удалите из введенного слова буквы, которые совпадают с его последней буквой.

22. Удалите из слова А все те буквы, которые хотя бы один раз попадают в слово Б.

23. Определите количество разных букв в слове.

24. Напишите программу подсчета количества встречающихся в тексте заданных фрагментов (из нескольких символов). К примеру, в предложении "банан упал на барабан" фрагмент "ан" встречается 3 раза.

25. Введены 2 слова. Напишите программу, которая определяет возможность составить из букв слова X слово Y.

26. Напишите программу, которая вычеркивает каждую третью букву из слова А.

27. Напишите программу, которая выясняет, на какую букву, гласную или согласную, оканчивается слово Y.

28. Напишите программу, которая вычисляет сумму позиций, на которых в данном слове А стоят буквы "р" и "с".

29. Напишите программу, которая шифрует текстовые сообщения. Можно воспользоваться следующим способом шифрования: задается ключ шифровки — который является целым числом и определяет величину на которую смещаются буквы русского алфавита, например: если ключ = 4, то в тексте буква "б" заменяется на "е" и т. д.

30. Напишите программу, которая дешифрует текстовое сообщение, зашифрованное с помощью программы из задачи 29.

Индивидуальное задание № 2

Создать программу по заданию своего варианта.

1. Необходимо выяснить, все ли буквы присутствуют среди последовательности вводимых символов, входящие в слово «привет».

2. Задан текст на русском языке, слова которого разделены точкой, запятой или пробелами. Слова, которые оканчиваются на -онок и -енок, необходимо представить во множественном числе.

3. С клавиатуры вводится строка. В случае, если она представляет собой запись римского числа, то необходимо преобразовать ее в целое число.

4. Составьте программу, которая позволяет перемещать курсор в рамках экрана вверх – вниз и влево – вправо, допускает ввод и выводе только символов русского алфавита независимо от переключения на русскую раскладку клавиатуры.

5. Введены 2 строки `s1` и `s2`. Необходимо определить, можно ли из `s1` получить строку `str2` путем перестановки символов. Составьте программу, которая решит данную задачу.

6. Составьте программу для сортировки строк в порядке, обратном алфавитному.

7. Составьте программу, которая вводит значения типа `string` и определяет, является ли каждое из них верным идентификатором, удовлетворяющим требованиям языка C++. При обнаружении ошибки написания идентификатора, следует вывести сообщение об этом.

8. Преобразуйте программу из задачи предыдущего варианта таким образом, чтобы она могла определять служебные слова языка C++ и запрещать попытки их использования. Для удобства уменьшите набор служебных слов, взяв только основные из них.

9. Введен текст, который состоит из последовательности строк. Признак окончания текста «0». Вывести на печать число содержащихся в каждой строке латинских и русских заглавных букв.

10. Дан текст, который представляет собой последовательность строк. Признак конца текста «0». Напечатать данный текст в порядке возрастания длины строки.

11. Текст представляет из себя последовательность строк. В каждой строке обязательно находятся фамилия, имя и отчество. Выведите все отчества в алфавитном порядке.

12. Введено восемь русских имен. Необходимо проверить, все ли введенные имена написаны с большой буквы. Если это не так, то изменить на прописную.

13. Введено слово. Необходимо определить, совпадают ли в нем третий и пятый символы. Вывести последний символ слова на экран. Подсчитать число символов в нем.

14. Введены 2 слова. Проверить условие, что 1 слово начинается на такую же букву, на которую оканчивается 2 слово? Подсчитать число символов в каждом слове. В обратном порядке осуществить вывод слов на экран.

15. Введено слово, которое состоит из четного количества букв. Необходимо вывести его вторую половину и в середине экрана вывести данное слово в обратном порядке.

16. Из заданного слова "программист" путем склеек и вырезок букв, составляющих слово, получить слова "рог" и "прога". Из полученных слов вывести те буквы, которые находятся на нечётных позициях.

17. Из слова "клоун" путем вырезок и склеек его букв получить слова "уклон", "кулон" и "колун". Из последних полученных трёх слов выбрать буквы, которые находятся на нечётных позициях.

18. Задано слово из нечетного количества букв. Поменять местами его половины, относительно одного символа. Вывести полученные слова в середине экрана в обратном порядке.

19. Задано слово из пятнадцати букв. Необходимо заменить местами его трети следующим образом: первую треть разместить на месте второй, вторую треть — на месте первой, третью — оставить. Удалить из получившегося слова каждый четвертый символ.

20. Задано слово `str1`. Образовать слово `str2`, из четных букв слова `str1`. Отобразить оба слова в обратном порядке.

21. Задано некоторое предложение. Вывести все буквы «а» и «о» из него и определить их число. Сколько слов содержится в предложении?

22. Заданы 2 предложения. Вывести те слова, которые встречаются в 2 предложениях только раз. Какое из предложений состоит из большего количества символов?

23. Задано некоторое предложение. Необходимо определить:

а) количество вхождений в него конкретного буквосочетания "ст";

б) количество вхождений в него указанного буквосочетания из 2 букв;

в) количество вхождений в него указанного произвольного буквосочетания.

24. Задано некоторое предложение. Все символы, находящиеся на нечетных позициях, поменять на букву “Ъ”. Отобразить в середине экрана последнее и первое слова предложения. Определить какое из этих слов длиннее другого?

3.5 Вопросы к защите лабораторной работы № 3

1) Что называется строкой?

2) Как описываются данные строкового типа?

3) Какой может быть максимальная длина строки? Как определить длину строки?

4) Каким образом указывается длина строки при описании строкового типа?

5) Перечислите основные операции над строками.

Список использованных источников

- 1 Грузина, Э.Э. Программирование. С++ : электронное учебное пособие [Электронный ресурс] / Э.Э. Грузина, К.С. Иванов, Л.В. Бондарева ; Министерство образования и науки РФ, Кемеровский государственный университет, Кафедра вычислительной математики. - Кемерово : Кемеровский государственный университет, 2015. - Ч. 2. - 120 с. - (ЭБС biblioclub.ru). - ISBN 978-5-8353-1604-5. - ISBN 978-5-8353-1851-3 (Ч.2). - URL: <http://biblioclub.ru/index.php?page=book&id=481536>
- 2 Белоцерковская, И.Е. Алгоритмизация. Введение в язык программирования С++ [Электронный ресурс] / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 197 с. - (ЭБС biblioclub.ru). - URL: <http://biblioclub.ru/index.php?page=book&id=428935>
- 3 Седжвик, Р. Алгоритмы на С++ [Электронный ресурс] / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 1773 с. - (ЭБС biblioclub.ru). - URL: <http://biblioclub.ru/index.php?page=book&id=429164>
- 4 Программирование на языке С++ в среде QtCreato [Электронный ресурс] / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк и др. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 716 с. (ЭБС biblioclub.ru). - URL: <http://biblioclub.ru/index.php?page=book&id=428929>
- 5 Колдаев, В.Д. Основы алгоритмизации и программирования [Электронный ресурс]: учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с. - (ЭБС znanium.com). - Режим доступа : <http://znanium.com/bookread2.php?book=484837>

Приложение А

(обязательное)

Критерии оценивания

Оценочные средства	Критерий для оценки «5»	Критерий для оценки «4»	Критерий для оценки «3»	Критерий для оценки «2»
Лабораторная работа	Лабораторная работа считается выполненной и оценивается в 5 баллов, если выполнены 95%-100% условий и требований, сформулированных в ней.	Лабораторная работа считается выполненной и оценивается в 4 балла, если выполнены 70%-94% условий и требований, сформулированных в ней.	Лабораторная работа считается выполненной и оценивается в 3 балла, если выполнены 40%-70% условий и требований, сформулированных в ней.	Лабораторная работа считается выполненной и оценивается в 1-2 балла, если выполнены менее 40% условий и требований, сформулированных в ней.

Приложение Б (рекомендуемое)

Лабораторная работа «Программирование с использованием одномерных массивов»

Цель работы: Научиться правильно описывать одномерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научиться решать задачи на использование массивов.

1 Ход работы

Индивидуальное задание № 1.

Вариант 0: Дан массив A_i размера N . Сформировать массив B_i элементы, которого вычисляются по формуле $B_i=(A_i)^2$. Отсортировать массив B_i по возрастанию.

2 Листинг программы

```
#include<iostream>
#include<ctime>
#include<cstdlib>
usingnamespacestd;

intmain()
{
    setlocale(0, "");
    int n;
    cout<<"длинамассива ";
    cin>> n;
    int* a = newint[n];
    int* b = newint[n];
    srand(time(0));
    for (inti = 0; i< n; i++)
    {
        a[i] = rand() % 201 - 100;
        b[i] = a[i] * 2;
    }
}
```

```

printf("массива:\n");
for (inti = 0; i< n; i++) printf("%5d", a[i]);
printf("\nмассив b:\n");
for (inti = 0; i< n; i++) printf("%5d", b[i]);
for (inti = 0; i< n - 1; i++)
    for (int j = i + 1; j < n; j++)
        if (b[i] > b[j])
            {
                inttmp = b[i];
                b[i] = b[j];
                b[j] = tmp;
            }
printf("\nотсортированный массив b:\n");
for (inti = 0; i< n; i++) printf("%5d", b[i]);
}

```

```

C:\> Консоль отладки Microsoft Visual Studio
длина массива 6
массив a:
  62  -65   18  -49   44  -66
массив b:
 124 -130   36  -98   88 -132
отсортированный массив b:
-132 -130  -98   36   88  124
C:\Users\Lena\source\repos\oplab4\Debug\oplab4.exe

```

Рисунок 1 – Вывод результатов работы программ

Вывод: Научились правильно описывать одномерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научиться решать задачи на использование массивов.

Приложение В (рекомендуемое)

Лабораторная работа «Программирование с использованием динамических двумерных массивов»

Цель работы: Научиться правильно описывать двумерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научиться решать задачи на использование массивов.

1 Ход работы

Индивидуальное задание № 1.

Вариант 0: Дана матрица размера $M \times N$. Найти в каждом столбце матрицы максимальный и минимальный элемент и поменять их местами.

2 Листинг программы

```
#include<iostream>
#include<ctime>
#include<cstdlib>
usingnamespacestd;

intmain()
{
    int n, m;
    cin>> m;
    cin>> n;
    int** mas = newint* [m];
    srand(time(0));
    for (inti = 0; i < m; i++)
    {
        mas[i] = newint[n];
        for (int j = 0; j < n; j++)
        {
            mas[i][j] = rand() % 21;
            printf("%4d", mas[i][j]);
        }
        cout<<endl;
    }
}
```

```

}
for (int j = 0; j < n; j++)
{
    int imin = 0, imax = 0;
    for (int i = 1; i < m; i++)
    {
        if (mas[i][j] < mas[imin][j]) imin = i;
        if (mas[i][j] > mas[imax][j]) imax = i;
    }
    if (imin != imax)
    {
        int tmp = mas[imin][j];
        mas[imin][j] = mas[imax][j];
        mas[imax][j] = tmp;
    }
}
cout << endl;
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++) printf("%4d", mas[i][j]);
    cout << endl;
}
for (int i = 0; i < m; i++) delete[] mas[i];
delete[] mas;
}

```

```

C:\> Консоль отладки Microsoft Visual Studio
4
6
 8   7  14  11  19  19
15  17  12  14  12   3
16  11   0   3  18  20
 8   7   7   3   5   6

16  17   0  11   5  19
15   7  12   3  12  20
 8  11  14  14  18   3
 8   7   7   3  19   6

```

Рисунок 1 – Вывод результатов работы программ

Вывод: научился правильно описывать двумерные массивы, заполнять массивы элементами, распечатывать содержимое массивов; научилась решать задачи на использование массивов.

Приложение Г (рекомендуемое)

Лабораторная работа «Программирование с использованием строк»

Цель работы: Получение навыков в организации ввода/вывода значений строковых типов данных. Получение практических навыков программирования задач с использованием строк.

1 Ход работы

Индивидуальное задание № 1:

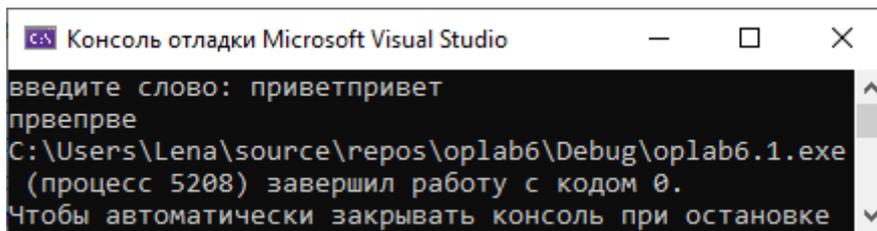
Составить алгоритм и написать программу согласно своему варианту. Можно использовать любой тип данных `char` или `string`

Вариант 0: Составьте программу, вычеркивающую каждую третью букву слова X.

Листинг программы

```
#include<iostream>
#include<string>
#include<Windows.h>
using namespace std;

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string x;
    printf("ВВЕДИТЕ СЛОВО: ");
    getline(cin, x);
    for (int i = x.length() / 3 * 3 - 1; i >= 0; i -= 3) x.erase(i, 1);
    printf("%s", x.c_str());
}
```



```
Консоль отладки Microsoft Visual Studio
введите слово: приветпривет
првепрве
C:\Users\Lena\source\repos\oplab6\Debug\oplab6.1.exe
(процесс 5208) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке
```

Рисунок 1 – Вывод результатов работы программ

Индивидуальное задание № 2:

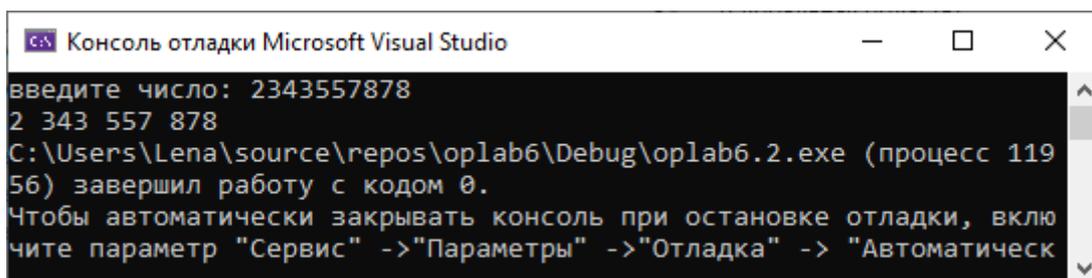
Составить алгоритм и написать программу согласно своему варианту. Можно использовать любой тип данных `char` или `string`.

Вариант 0: Дано натуральное число. Получить строку, в которой тройки цифр этого числа разделены пробелом, начиная с правого конца. Например, число 1234567 преобразуется в 1 234 567.

Листинг программы

```
#include<iostream>
#include<string>
#include<Windows.h>
usingnamespacestd;

intmain()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string s;
    printf("введитечисло: ");
    getline(cin, s);
    for (inti = s.length()-3; i> 0; i -= 3) s.insert(i, " ");
    printf("%s", s.c_str());
}
```



```
Консоль отладки Microsoft Visual Studio
введите число: 2343557878
2 343 557 878
C:\Users\Lena\source\repos\oplab6\Debug\oplab6.2.exe (процесс 119
56) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, вклю
чите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматическ
```

Рисунок 1 – Вывод результатов работы программ

Вывод: Получили навыки в организации ввода/вывода значений строковых типов данных. Получила практические навыки программирования задач с использованием строк.