

Лабораторная работа №3

студента группы ПИ-212

Гаркуши Артема Андреевича

Выполнение _____

Защита _____

ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ. ФАЙЛОВЫЙ ВВОД-ВЫВОД.

Цель работы: ознакомиться с организацией двумерных массивов в языке C/C++;
приобрести практические навыки в файловом вводе-выводе данных;

СОДЕРЖАНИЕ РАБОТЫ

Выбрать алгоритм, составить его блок-схему и программу для решения выбранного варианта задания.

Программа должна запрашивать имена входного и выходного файлов. Программа должна осуществлять ввод исходной матрицы A из файла. Матрица A – матрица вещественных чисел.

Программа должна вывести исходную матрицу A на экран и результирующий массив X на экран и в выходной файл. Вывод матрицы A должен осуществляться в виде квадратной матрицы, вывод массива X в виде строки.

Вариант №1

1. Дана матрица $A(5 \times 5)$. Определить массив X из 5 элементов, каждый из которых равен сумме чётных элементов соответствующей строки верхней треугольной матрицы.

Ход работы:

Текст программы:

```
#include <fstream>
#include <iostream>
#include <locale.h>
#define N 5
using namespace std;

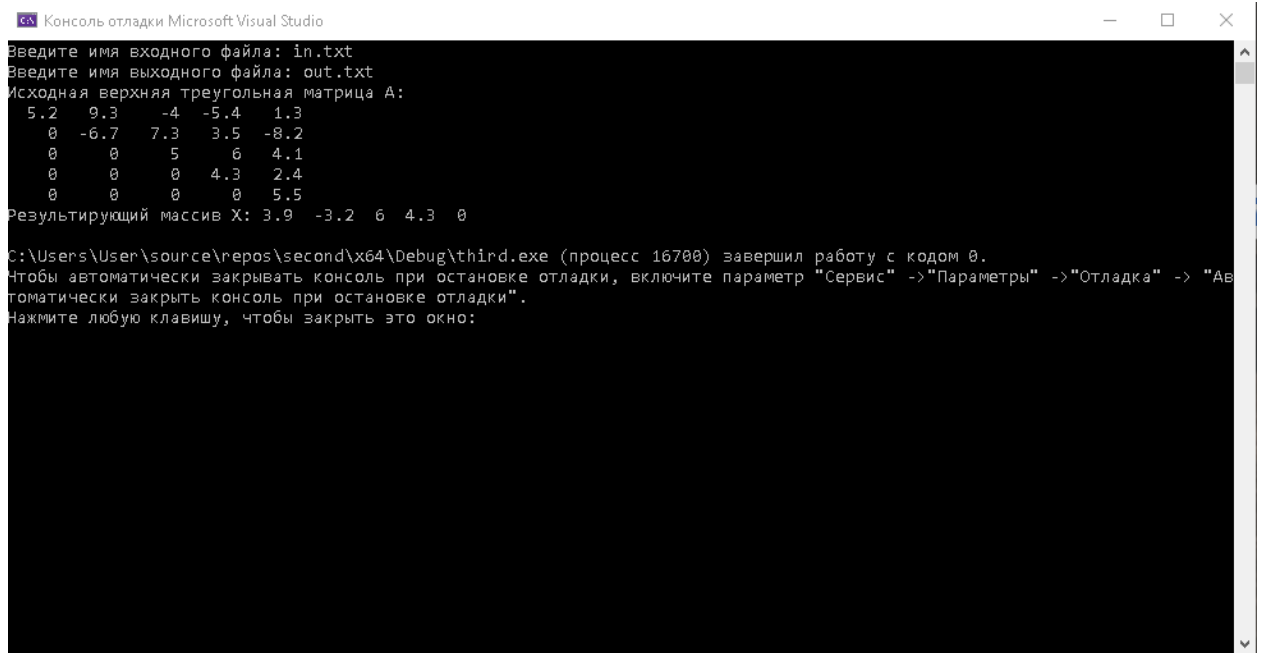
int main()
{
    setlocale(LC_CTYPE, "");
    int i, j, z = 0;
    float mas[N][N], x[N], q;
    string infile, outfile;
    cout << "Введите имя входного файла: ";
    cin >> infile;
    cout << "Введите имя выходного файла: ";
    cin >> outfile;

    ifstream fin(infile);
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            fin >> mas[i][j];
    fin.close();

    cout << "Исходная верхняя треугольная матрица A: " << "\n";
    for (int i = 0; i < N; ++i, ++z)
    {
        for (int j = 0; j < N; ++j)
        {
            cout.width(5);
            cout << mas[i][j] << " ";
        }
        cout << endl;
        x[z] = mas[i][1] + mas[i][3];
    }
    cout << "Результирующий массив X: ";
    for (int i = 0; i < N; ++i)
        cout << x[i] << " ";
    cout << endl;

    ofstream fout(outfile);
    fout << "Результирующий массив X: ";
    for (int i = 0; i < N; ++i)
        fout << x[i] << " ";
    fout.close();
}
```

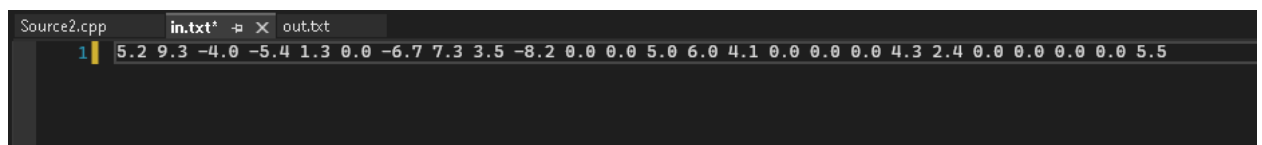
Результат выполнения программы:



```
Консоль отладки Microsoft Visual Studio
Введите имя входного файла: in.txt
Введите имя выходного файла: out.txt
Исходная верхняя треугольная матрица A:
5.2  9.3  -4  -5.4  1.3
0  -6.7  7.3  3.5  -8.2
0  0  5  6  4.1
0  0  0  4.3  2.4
0  0  0  0  5.5
Результирующий массив X: 3.9 -3.2 6 4.3 0

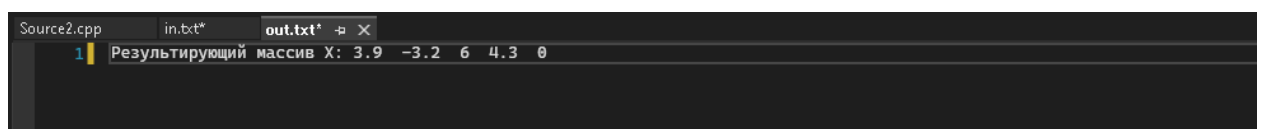
C:\Users\User\source\repos\second\x64\Debug\third.exe (процесс 16700) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рис. 1. Выполнение программы



```
Source2.cpp  in.txt*  out.txt
1 | 5.2 9.3 -4.0 -5.4 1.3 0.0 -6.7 7.3 3.5 -8.2 0.0 0.0 5.0 6.0 4.1 0.0 0.0 0.0 4.3 2.4 0.0 0.0 0.0 0.0 5.5
```

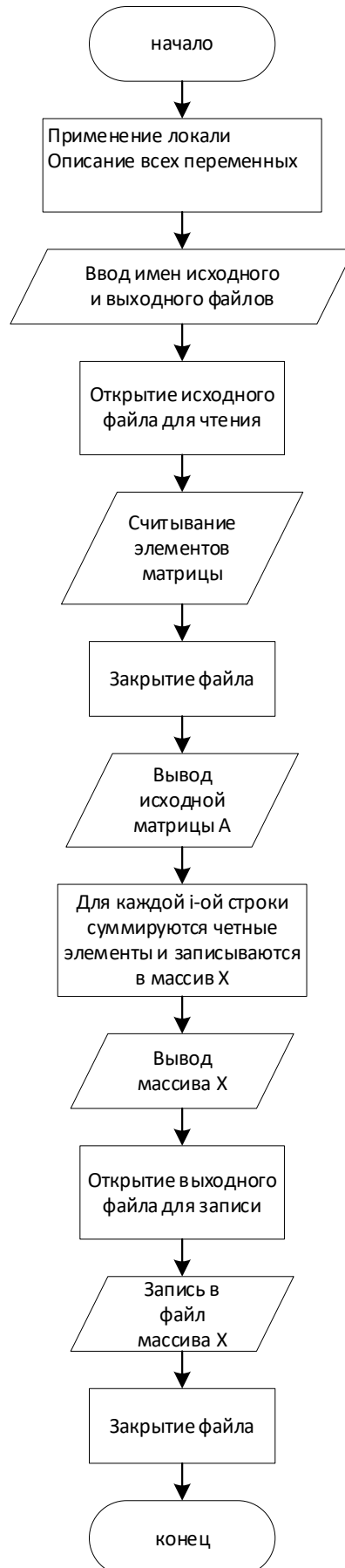
Рис. 2. Исходный файл



```
Source2.cpp  in.txt*  out.txt*
1 | Результирующий массив X: 3.9 -3.2 6 4.3 0
```

Рис. 3. Выходной файл

Блок-схема:



Ответы на контрольные вопросы:

1. Каковы в языке C/C++ принципы размещения в памяти многомерных массивов? Как производится их описание?

Массив располагается в непрерывном отрезке памяти, под каждый элемент массива выделяется `sizeof(T)` байт, соответственно размер памяти, необходимой для размещения всего массива, равен `N*sizeof(T)` байт. Эта величина ограничена сверху платформой и компилятором.

В языке C/C++ многомерные (в частности двумерные) массивы – это массивы, элементами которых в свою очередь являются другие массивы.

Например, конструкция:

```
int mas[3][5];
```

описывает массив из трех элементов, каждым из которых является массив из пяти элементов целого типа. Фактически это матрица (3×5). Доступ к элементам осуществляется указанием двух индексов:

```
mas[0][1]=25; // присвоить 25 элементу, находящемуся в 1-й строке и 2-м столбце
```

2. Как производится обращение к элементам многомерного массива?

```
mas[0][1]=25;
```

3. Какими способами можно произвести заполнение многомерного массива элементами?

Ввод элементов массива с клавиатуры:

```
int a[5][5], i, j;
```

```
for(i=0; i<5; i++)
```

```
for(j=0; j<5; j++)
```

```
scanf("%d", &a[i][j]);
```

Двумерные массивы можно инициализировать при описании так же как и одномерные. Например:

```
int a[2][3] = { { 1, 2, 3 }, { 4, 5, 6 } }
```

4. Как осуществляется файловый ввод-вывод в языке C/C++?

Ввод-вывод на верхнем уровне в языке Си осуществляется через потоки. Поток является файлом или физическим устройством (например, принтером или монитором), которое управляется с помощью указателей на объект типа `FILE` (определенный в `stdio.h`).

Для организации потокового файлового ввода-вывода в языке C++ необходимо подключить заголовочный файл `<fstream>`. В `<fstream>` определены несколько классов и подключены заголовочные файлы `<ifstream>` – файловый ввод и `<ofstream>` – файловый вывод.

5. В каком файле определены прототипы функций ввода-вывода верхнего уровня?

Прототипы всех функций ввода/вывода верхнего уровня содержатся в файле `stdio.h`.

6. Какие функции осуществляют открытие и закрытие файла?

В Си открывается поток с помощью функции fopen:

```
FILE *fopen(char *pathname, char *type);
```

Для закрытия потока используется функция fclose:

```
int fclose(FILE *fp);
```

В C++ :

```
ifstream fin(infile);
```

```
fin.close();
```

7. Какие функции предназначены для форматированного ввода- вывода данных?

cout.width(5); - ширина 5

cout << endl; - конец строки