

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра систем автоматизированного проектирования**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Программирование»**  
**Тема: Файлы для ввода-вывода данных, двумерные массивы.**  
**Функции**

Студентка гр. 3353

Карпенко А.Ю

Преподаватель

Калмычков В.А

Санкт-Петербург

2023

## ОГЛАВЛЕНИЕ:

Исходная формулировка задания	
Цели работы	
Математическая постановка задачи	
Контрольный пример	
Особенности выполнения на компьютере	
Реализация ввода/вывода	
Организация хранения данных	
Представление алгоритма решения задачи. Блок-схема	
Текст программы	
Результаты работы программы	
Вывод	

### Исходная формулировка задания

Найти наибольший элемент заштрихованной области таблицы A размера N x N

г) m

	X					
		X				
X	X	X	X			k
				X		
					X	
X	X	X	X	X	X	X

### Цели работы

- изучение работы с двумерными массивами.
- изучение работы с функциями
- закрепление работы fstream
- закрепление условных конструкций(ветвления) и циклов

### Математическая постановка задания

Дано: максимальный размер матрицы, координата элемента отсчета по горизонтали, координата элемента отсчета по вертикали, двумерный массив

Найти: максимальный элемент согласно следующему алгоритму

Алгоритм:

От элемента отсчета берется диагональ, строчка влево и полностью строчка в момент, когда диагональ заканчивается.

Если выполняется одно из условий, то элемент с координатами (i,j) берется в рассмотрение.

- $i==k \ \&\& \ j<m$
- $j==(i+1)$
- $(i+1)==(n-1)$

### Контрольный пример

анализируемая матрица

1	2	3	4	5
6	7	8	9	10
111	12	13	14	155
16	17	18	19	20
21	22	23	24	255.5

анализируемые значения

1				
	7			
111	12	13		
			19	
21	22	23	24	255.5

ответ

255.5

### Особенности выполнения на компьютере

Тип float представляет вещественное число с плавающей точкой в диапазоне от  $-3.4E-38$  до  $3.4E38$  и в памяти занимает 4 байта (32 бита), что накладывает свои ограничения на величину  $m$  и на элементы массива. Тип переменной int, который хранит целочисленные значения в диапазоне от  $-32768$  до  $32767$  и занимает 2 байта (16 бит) накладывает свои ограничения на  $x$ ,  $m$ ,  $k$ . В программе применяется двумерный динамический массив

Объявление и выделение памяти:

```
float **A = new float*[n];
for (int i = 0; i < n; ++i) {
    A[i] = new float[n];
}
int **pA;
```

Обращение к элементу массива:

$A[\text{номер строки}][\text{номер столбца}]$

Освобождение памяти:

```
for (int i = 0; i < n; i++)
{
    delete[] A[i];
}
delete[] A;
```

В процессе выполнения программы потребуется реализация условных конструкций:

```
if (условие 1) {(выполняется если условие 1 истина);}
else if(условие 2) {выполняется если условие 2 истина;}
else {выполняется если условие 1 И условие 2 ложь;;}
```

А так же потребуются циклы, будем использовать циклы:

```
while(условие провидения целиа)
{
    (тело цикла)
}
for(начальные условия; условия продолжения; действие после завершения одного тела)
{
    (тело цикла)
}
```

### Организация ввода-вывода

- 1) Программа представляется в выходной файл вывод исходную формулировку задания

**Представление и задание:**

*Карпенко Анастасия*

*группа 3353*

*дата создания: 5.11.23*

*номер задания: 4т*

- 2) **Чтение массива из файла:**

Организовано путем взятия каждого элемента функцией оператором извлечения >>

В первую очередь снимаются элементы в первых трех строках, несущие желаемый размер массива, параметры  $m$  и  $k$ .

**формат:**

ddd

ddd

ddd

Потом через алгоритм анализа элементов, снимается значения элементов массива, начиная с четвертой строки и в последующих в двумерный динамический массив

**формат:**

ddd.dd<sub>11</sub>ddd.dd<sub>12</sub>...ddd.dd<sub>1n</sub>  
 ddd.dd<sub>21</sub>ddd.dd<sub>22</sub>...ddd.dd<sub>2n</sub>

...  
 ddd.dd<sub>n1</sub>ddd.dd<sub>n2</sub>...ddd.dd<sub>nn</sub>

**Ожидаема формат файла:**

ddd

ddd

ddd

ddd.dd<sub>11</sub>ddd.dd<sub>12</sub>...ddd.dd<sub>1n</sub>

ddd.dd<sub>21</sub>ddd.dd<sub>22</sub>...ddd.dd<sub>2n</sub>

...  
 ddd.dd<sub>n1</sub>ddd.dd<sub>n2</sub>...ddd.dd<sub>nn</sub>

**3) Помещение результата в файл:****Вывод исходного массива**

*анализируемая матрица: \*выводит элементы двумерного массива в матричном виде\**

ddd.dd<sub>11</sub>ddd.dd<sub>12</sub>...ddd.dd<sub>1n</sub>

ddd.dd<sub>21</sub>ddd.dd<sub>22</sub>...ddd.dd<sub>2n</sub>

...  
 ddd.dd<sub>n1</sub>ddd.dd<sub>n2</sub>...ddd.dd<sub>nn</sub>

**Вывод обрабатываемых значений**

*анализируемые значения: выводит элементы, среди которых будет искажаться максимальный.*

ddd.dd<sub>11</sub>ddd.dd<sub>12</sub>...ddd.dd<sub>1n</sub>

ddd.dd<sub>21</sub>ddd.dd<sub>22</sub>...ddd.dd<sub>2n</sub>

...  
 ddd.dd<sub>21</sub>ddd.dd<sub>22</sub>...ddd.dd<sub>2n</sub>

...  
 ddd.dd<sub>n1</sub>ddd.dd<sub>n2</sub>...ddd.dd<sub>nn</sub>

**Вывод результата:**

ddd.dd

Для ввода из файла используем библиотеку fstream

(название потока)>>(имя переменной);

Для вывода в файл используем библиотеку fstream

(название потока)<<«Текст»<<(имя переменной);

Так же стоит упомянуть, что для работы с файлами потребуется открывать потоки вывода и открыть файлы, а так же :

fstream (название потока) ;

(название потока).open(«Имя файла», ios::in для ввода и out для вывода));

(название потока).close();

Файл обязательно открыть в режиме БЕЗ пропуска пробелов и символов перехода на новую строку

f.unsetf(ios::skipws);

## Организация хранения данных

### Глобальные переменные

Название переменной	Тип переменной	пояснение
n	const int	Хранит размер обрабатываемого массива
**matrix	float	Обрабатываемый двумерный массив
x	int	Хранит размер вводного двумерного массива
m	int	Хранит координату элемента по горизонтали
k	int	Хранит координату элемента по вертикали
maxim	float	Хранит максимальное значение

### Локальные переменные

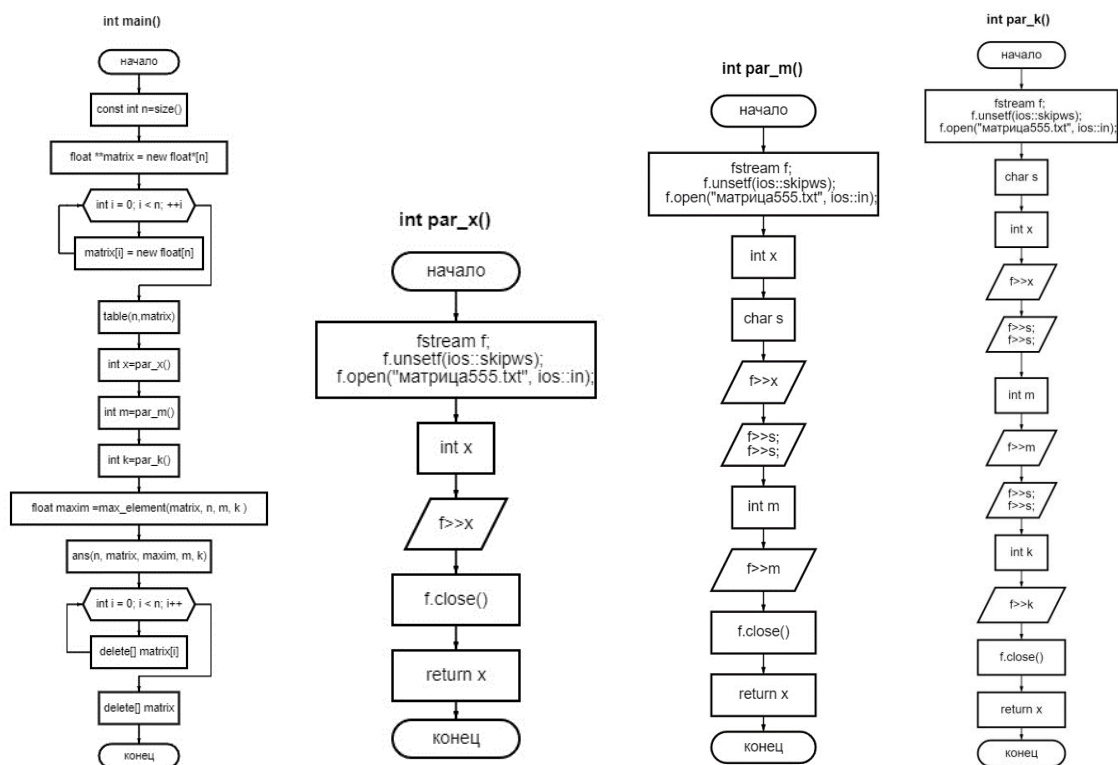
Название переменной	Тип переменной	пояснение
s	char	Хранит взятый из файла символ
i, j	unsigned	Переменные счетчика
l	int	Координата по горизонтали элемента — начала диагонали

## Представление алгоритма решения задачи. Блок-схема

Программа считывает желаемый размер массива из файла, координаты элемента отсчета по горизонтали и вертикали и сам массив, потом выводит в выходной файл представление, анализируемую матрицу, анализируемые значения и ответ или сообщение о том, что входной двумерный массив анализу не подлежит.

Имя функции	Назначение	Параметры		Возвращаемые значения	Внешние эффекты
		входные	выходные		
size	Выводит размер матрицы, которую необходимо проанализировать	-	int n	Int (возвращает размер матрицы)	Открытие и закрытие файла
par_x	Выводит максимальный размер матрицы	-	int x	Int (возвращает максимальный размер двумерного массива)	Открытие и закрытие файла
par_m	Выводит значение параметра m	-	int m	Int (возвращает координату точки отсчета по горизонтали)	Открытие и закрытие файла
par_k	Выводит значение параметра k	-	int k	Int (возвращает координату точки отсчета по горизонтали)	Открытие и закрытие файла
table	Считывает матрицу необходимого размера	int n, float** matrix	-	-	Открытие и закрытие файла. Считывание матрицы необходимого размера

max_element	Выводит максимальный элемент	float** matrix, int n, int m, int k	float maxim	Float (возвращает максимальный элемент)	-
outfile	Выводит строку обрабатываемой матрицы	float* matrix, int n, ofstream &res	-	-	Выводит в файл строку исходного массива.
ans	Выводит приветствие, матрицу, анализируемые значения, ответ или сообщение об ошибке	int n, float** matrix, float maxim, int m, int k	-	-	Создает выходной файл, выводит обрабатываемую матрицу, анализируемые значения и ответ.





```

        if(i>=n) {n=i; break;}
    }
    f.close();

    if (n<=x) {return n;}
    else if (n>x) {return x;}
    else {return 0;}
}

int par_x(){
    fstream f;
    f.unsetf(ios::skipws);
    f.open("матрица555.txt", ios::in);
    int x;
    f>>x;
    f.close();
    return x;
}

int par_m(){
    fstream f;
    f.unsetf(ios::skipws);
    f.open("матрица555.txt", ios::in);
    char s;
    int x;
    f>>x;
    f>>s;
    f>>s;
    int m;
    f>>m;
    f.close();
    return m;
}

int par_k(){
    fstream f;
    f.unsetf(ios::skipws);
    char s;
    f.open("матрица555.txt", ios::in);
    int x;
    f>>x;
    f>>s;
    f>>s;
    int m;
    f>>m;
    f>>s;
    f>>s;
    int k;
    f>>k;
    f.close();
    return k;
}

void table(int n, float** matrix){
    fstream f;
    f.unsetf(ios::skipws);
    f.open("матрица555.txt", ios::in);
    char s;
    int x;
    f>>x;
    f>>s;
    f>>s;
    int m;
    f>>m;
    f>>s;
    f>>s;
    int k;
    f>>k;
    f>>s;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            f>>matrix[i][j];
            f>>s;
            if(j==(n-1) && !f.eof()) {while(1) {if(s=='\n') break; f>>s;}}
        }
    }
    f.close();
}

float max_element(float** matrix, int n, int m, int k ) {
    int l=m-k;
    float maxim=0;
    for (int i=0;i<n;i++){
        for (int j=0;j<n;j++){
            if (i==k && j<m){maxim=max(maxim,matrix[i][j]);}
            else if (j==(i+1)) {maxim=max(maxim,matrix[i][j]);}
            else if ((i+1)==(n-1) ) {maxim=max(maxim,matrix[i][j]);}
        }
    }
    return maxim;
}

void outfile(float* matrix, int n,ofstream &res)
{
    unsigned j;
    for(int j=0; j<n;j++){
        res<<setw(4)<<matrix[j]<<setw(4)<<" ";
    }
    res<<endl;
}

void ans(int n, float** matrix, float maxim, int m, int k){

```



```

ofstream res("результат555.txt");
res<<"Карпенко Анастасия\n";
res<<"группа 3353\n";
res<<"дата создания: 5.11.23\n";
res<<"номер задания: 4т\n";
res<<"анализируемая матрица\n";
for(int i=0; i < n;i++){
    outfile(matrix[i], n, res);
    res<<"\n";
}
res<<"анализируемые значения\n";
int l=m-k;
for (int i=0;i<n;i++){
    for (int j=0;j<n;j++){
        if (i==k && j<m){res<<setw(5)<<matrix[i][j]<<" ";}
        else if (j==(i+l)) {res<<setw(5)<<matrix[i][j]<<setw(5)<<" ";}
        else if ((i+l)==(n-1) ) {res<<setw(5)<<matrix[i][j]<<" ";}
        else {res<<setw(6)<<" ";}
    }
    res<<"\n";
}
res<<"ответ\n";
if (maxim==0){res<<"таблица не подлежит анализу\n";}
else {res<<maxim<<"\n";}
res.close();
}

int main(){
    const int n=size();
    float **matrix = new float*[n];
    for (int i = 0; i < n; ++i) {
        matrix[i] = new float[n];
    }
    table(n,matrix);
    int x=par_x();
    int m=par_m();
    int k=par_k();
    float maxim;
    maxim=max_element(matrix, n, m, k );
    ans(n, matrix, maxim, m, k);
    for (int i = 0; i < n; i++) {delete[] matrix[i];}
    delete[] matrix;
}

```

## Результаты работы программы

Входной файл	Результат работы
<pre> 1  4 2  2 3  2 4  1 2 3 4 5 5  6 7 8 9 10 6  111 12 13 14 155 7  16 17 18 19 20 8  21 22 23 24 255.5 </pre>	<pre> 5  анализируемая матрица 6      1      2      3      4 7      6      7      8      9 8     111     12     13     14 9     16     17     18     19 10 анализируемые значения 11      1 12             7 13     111     12     13 14     16     17     18     19 15 ответ 16 111 </pre>
<pre> 1  7 2  2 3  2 4  1 2 3 4 5 5  6 7 8 9 10 6  111 12 13 14 155 7  16 17 18 19 20 8  21 22 23 24 255.5 </pre>	<pre> 5  анализируемая матрица 6      1      2      3      4      5 7      6      7      8      9     10 8     111     12     13     14     155 9     16     17     18     19     20 10     21     22     23     24 255.5 11 анализируемые значения 12      1 13             7 14     111     12     13 15                           19 16     21     22     23     24 255.5 17 ответ 18 255.5 </pre>

<pre> 1 7 2 2 3 2 4 1 2 3 5 6 7 8 9 10 6 111 12 13 14 155 7 16 17 18 19 20 8 21 22 23 24 255.5 </pre>		<pre> 5 анализируемая матрица 6 1 2 3 7 6 7 8 8 111 12 13 9 анализируемые значения 10 1 11   7 12 111 12 13 13 ответ 14 111 15 </pre>	
<pre> 1 4 2 5 3 5 4 1 2 3 4 5 5 6 7 8 9 10 6 111 12 13 14 155 7 16 17 18 19 20 8 21 22 23 24 255.5 </pre>		<pre> 5 анализируемая матрица 6 1 2 3 4 7 6 7 8 9 8 111 12 13 14 9 16 17 18 19 10 анализируемые значения 11 1 12   7 13   13 14 16 17 18 19 15 ответ 16 19 </pre>	

### Вывод

На основе работы изучили работу с двумерными массивами и функциями, закрепили знания работы с циклами, ветвлением и библиотекой stream