*Констатные переменные:*

*Целочисленные:*

**N\_MAX =** 15;//максимальное число строк в матрице

**M\_MAX** = 15**;** //максимальное число столбцов в матрице

*Символьные:*

**FILENAME\_1** ="tests\_1/src12.txt"; //путь к исходным файлам для 1 матрицы

**FILENAME\_2** = "tests\_2/src12.txt"; //путь к исходным файлам для 2 матрицы

**НАЧАЛО ОСНОВНОГО АЛГОРИТМА**

*Целочисленные переменные:*

**N, M – размеры матрицы**

**CodeError** – код ошибки

**ARRAY\_1[N\_MAX][M\_MAX]** - объявление первой матрицы

**ARRAY\_2[N\_MAX][M\_MAX]** - объявление второй матрицы

Вызов  **CHECK\_ERRORS(ARRAY\_1, N, M, CodeError, FILENAME\_1)**

Вызов  **GetCode(CodeError)**

**НАЧАЛО ЕСЛИ**

**Вывод:** *Исходные данные первого массива:*

**Вызов**  PrintArray(ARRAY\_1, N, M)

**Вызов**  SwapArray(ARRAY\_1, N, M)

**Вывод:** *Массив после перестановки:*

**Вызов**  PrintArray(ARRAY\_1, N, M)

**Вызов**  MaxArray(ARRAY\_1, N, M)

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ОСНОВНОГО АЛГОРИТМА**

**АЛГОРИТМ**  CHECK\_ERRORS(int ARRAY[][M\_MAX], int& N, int& M, int& CodeError, const char\* FILENAME)

**НАЧАЛО АЛГОРИТМА**

*Целочисленные переменные:*

**CntVec = 0 -** счетчик реального размера массива

**CntLine = 0 -** счетчик для ввода

**CntColumn = 0 -** счетчик для ввода

**buff -** буффер для проверки корректности вводимых данных

**НАЧАЛО ЕСЛИ** файл пуст

CodeError = -1

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** файл пуст **И** CodeError == 0

CodeError = -2

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = 0

**НАЧАЛО ЦИКЛА ПОКА** символы не закончились

**Ввод** очередного символа в буффер

**НАЧАЛО ЕСЛИ** очередной символ не цифра

CodeError = -3  
 **КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = 0

Ввод **N**

Ввод **M**

**НАЧАЛО ЦИКЛА ПОКА** символы не закончились

**Ввод** очередного символа в буффер

CntVec++

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЕСЛИ** CntVec > N\*M

CodeError = -4  
 **КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CntVec < N\*M

CodeError = -5  
 **КОНЕЦ ЕСЛИ**  
**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = 0

**Ввод** buff, buff

**НАЧАЛО ЦИКЛА ПОКА** символы не закончились

**Ввод** очередного символа в ARRAY[CntLine][CntColumn]

CntColumn++

**НАЧАЛО ЕСЛИ** CntColumn == M

CntLine++

CntColumn = 0

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЕСЛИ**

**КОНЕЦ АЛГОРИТМА**

**АЛГОРИТМ** GetCode(int CodeError)

**НАЧАЛО АЛГОРИТМА**

**Ввод** CodeError

**НАЧАЛО ЕСЛИ** CodeError = -1

**Вывод:** *Ошибка! файл не найден*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = -2

***Вывод:*** *Ошибка! файл пуст!*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = -3

***Вывод:*** *Ошибка! Допустим ввод только цифр!*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = -4

**Вывод:** *Ошибка! Количество входных элементов больше размера массива!*

**КОНЕЦ ЕСЛИ**

**НАЧАЛО ЕСЛИ** CodeError = -5

**Вывод:** *Ошибка! Количество входных элементов меньше размера массива!*

**КОНЕЦ ЕСЛИ**

**КОНЕЦ АЛГОРИТМА**

**АЛГОРИТМ** SwapArray(int ARRAY[][M\_MAX], int N, int M)

**НАЧАЛО АЛГОРИТМА**

*Целочисленные переменные:*

**buff -** буффер для обмена

**НАЧАЛО ЦИКЛА** по **i** до **N**

i++

**НАЧАЛО ЦИКЛА** по **j** до **M/2**

j++

buff = ARRAY[i][j]

ARRAY[i][j] = ARRAY[i][M - 1 - j]

ARRAY[i][M - 1 - j] = buff

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ АЛГОРИТМА**

**АЛГОРИТМ** MaxArray(int ARRAY[][M\_MAX], int N, int M)

**НАЧАЛО АЛГОРИТМА**

*Целочисленная переменная:*

**max** = 0; //т.к. просят найти максимум среди положительных чисел ставим изначальное значение максимума = 0

**НАЧАЛО ЦИКЛА** по **i** до **N**

i++

**НАЧАЛО ЦИКЛА** по **j** до **M**

j++

**НАЧАЛО ЕСЛИ** ARRAY[i][j] > max

max = ARRAY[i][j]

**КОНЕЦ ЕСЛИ**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЦИКЛА**

**НАЧАЛО ЕСЛИ** max == 0

**Вывод:** Все элементы отрицательны! Максимума нет!

**НАЧАЛО ИНАЧЕ**

**Вывод:** Максимум: max

**КОНЕЦ ИНАЧЕ**

**КОНЕЦ ЕСЛИ**

**КОНЕЦ АЛГОРИТМА**

**АЛГОРИТМ** PrintArray(int ARRAY[][M\_MAX], int N, int M)

**НАЧАЛО АЛГОРИТМА**

**НАЧАЛО ЦИКЛА** по **i** до **N**

i++

**НАЧАЛО ЦИКЛА** по **j** до **M**

j++

**Вывод:** ARRAY[i][j]

**КОНЕЦ ЦИКЛА**

**КОНЕЦ ЦИКЛА**

**КОНЕЦ АЛГОРИТМА**