## Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**ДИНАМИЧЕСКИЕ ПЕРЕМЕННЫЕ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ С**

## отчет о лабораторной работе № 12

по дисциплине

*ПРОГРАММИРОВАНИЕ*

***ВАРИАНТ 10***

|  |  |  |
| --- | --- | --- |
| Выполнил: | ст. гр. 220451 | Курбаков М.Ю. |
| Проверил: | асс. каф. ИБ | Курбаков М.Ю. |

## Тула, 2016 г.

**ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель:** научиться использовать динамические переменные.

**Задача:** в данной работе требуется написать программу с использованием динамических переменных, в которой данные сначала из файла считываются в память, потом обрабатываются и записываются в файл.

**ЗАДАНИЕ НА РАБОТУ**

Выполните задание по работе с двумерными массивами из лабораторной работы № 4 «Работа с массивами в языке программирования C», и используя динамическое создание массивов. Размер массива должен задаваться пользователем с клавиатуры. Результат работы программы должен выводиться на экран и в файл.

**Вариант задания по двумерным массивам:** Переставляя строки и столбцы матрицы, добиться, что бы в левом верхнем углу оказался наибольший элемент матрицы (один из них).

# СХЕМА АЛГОРИТМА

## Схема алгоритма для перемещения наибольшего элемента матрицы в левый верхний угол, путем перестановки сток и столбцов, представлена на рисунке 1.

Начало

Локальные параметры:

Input, Output – имена файлов для ввода и вывода соответственно; n, m – размер матрицы;

mas – матрица (n×m);

Количество строк (n)

нет

Данные введены верно?

да

Количество столбцов (m)

нет

Данные введены верно?

да

Input := name\_for\_input()

Матрица из файла

С Стр. 6

Рисунок 1 – Схема алгоритма для перемещения наибольшего элемента матрицы в

левый верхний угол

С Стр. 5

mas := processing(mas, n, m)

Output := name\_for\_output()

Outputf(Output, m, n, mas)

Конец

Рисунок 1 – Схема алгоритма для перемещения наибольшего элемента матрицы в левый верхний угол (продолжение)

Схема алгоритма для ввода названия конечного файла представлена на рисунке 2.

Вход

FILE\* name\_for\_output() – подпрограмма для ввода и проверки имени файла, в который нужно вывести результат

Имя файал

нет

Файл можно открыть для записи?

да

Выход

Рисунок 2 – Схема алгоритма для ввода названия коечного файла

Схема алгоритма для вывода результата на рисунке 3.



Вход

void Outputf(FILE \*NEW, int m, int n, int \*\*mas) – подпрограмма вывода результата в файл и на экран

i := 1, i <= n, 1

j := 1, j <= m, 1

mas[i][j] матрицы

Вывод на экран и вывод в файл

j

i

Выход

Рисунок 3 – Схема алгоритма для вывода результата

**ТЕКСТ ПРОГРАММЫ**

Текст программы перемещения наибольшего элемента матрицы в левый верхний угол, путем перестановки сток и столбцов, на языке программирования Си представлен в листинге 1.

**Листинг 1. Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <math.h>

#include <conio.h>

#include <malloc.h>

FILE\* name\_for\_output();

float\*\* getmem(float \*\*arr, int n, int m); float\*\* freemem(float \*\*arr, int n);

float\*\* processing(float\*\* mas, int n, int m);

void Outputf(FILE \*NEW, int m, int n, float \*\*mas); void main()

{

FILE \*Output; float \*\*mas = NULL; int n = 0, m = 0; int s = 0; setlocale(LC\_ALL, "Russian");

do

{

printf("Введите количество строк в матрице - n, где n > 2: ");

scanf("%i", &n);

} while (n < 2);

do

{

printf("Введите количество столбцов в матрице- m, где m > 2: ");

scanf("%i", &m);

} while (m < 2);

mas = getmem(mas, n, m); if (mas != NULL)

{

printf("Хотите заполнить матрицу вручную, введите 1. "

" Иначе матрица заполнится случайными числами. >> "); scanf("%i", &s);

if (s == 1)

{

for (int i = 0; i < n; i++) for (int j = 0; j < m; j++)

{

}

else

{

printf("Элемент матрицы[%i][%i]: ", i + 1, j + 1); scanf("%f", &mas[i][j]);

}

int r, ver = 0;

printf("\nВведите диапазон генератора случайных чисел: ");

ver = scanf("%d", &r);

for (int i = 0; i < n; i++)

for (int j = 0; j <= m - 1; j++) mas[i][j] = rand() %r;

}

printf("Ваша матрица:\n"); for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

printf("%8.2f", mas[i][j]);

}

printf("\n");

}

mas = processing(mas, n, m);

Output = name\_for\_output(); Outputf(Output, m, n, mas);

freemem(mas, n);

printf("Программа завершена, результат записан в файл.\n");

}

}

FILE\* name\_for\_input()

{

char Ch[125]; FILE \*f;

do

{

printf("Введите название файла, из которого необходимо считать" " информацию: ");

scanf("%s", Ch);

f = fopen(Ch, "rt");

} while (f == fopen(Ch, "rt") != NULL); return f;

}

FILE\* name\_for\_output()

{

FILE \*f;

char Ch[125]; do

{

printf("\nВведите назание файла, В который необходимо записать" " результат: ");

scanf("%s", Ch);

} while ((f = fopen(Ch, "wt")) == NULL); return f;

}

float\*\* getmem(float \*\*arr, int n, int m)

{

arr = NULL;

arr = (float\*\*)calloc(n, sizeof(float\*)); if (arr == NULL)

{

printf("Ошибка при распределении памяти \n"); return NULL;

}

for (int i = 0; i < n; i++)

{

arr[i] = NULL;

arr[i] = (float\*)calloc(m, sizeof(float)); if (arr[i] == NULL)

{

printf("Ошибка при распределении памяти \n"); return NULL;

}

}

return arr;

}

float\*\* freemem(float \*\*arr, int n)

{

for (int j = 0; j < n; j++)

{

free(arr[j]); arr[j] == NULL;

}

free(arr); arr == NULL; return arr;

}

float\*\* processing(float\*\* mas, int n, int m)

{

float max = mas[0][0]; int I = 0, J = 0;

for (int i = 0; i < n; i++) for (int j = 0; j < m; j++)

if (max < mas[i][j])

{

max = mas[i][j]; I = i;

J = j;

}

if ((I == 0) && (J == 0))

printf("Максимальный элемент уже находится в верхнем левом углу!");

else

{

//Певая строка меняется с I строкой

for (int j = 0; j < m; j++)

{

float z = mas[0][j]; mas[0][j] = mas[I][j]; mas[I][j] = z;

}

//Первый столбец меняется с J столбцом

for (int i = 0; i < n; i++)

{

float z = mas[i][0]; mas[i][0] = mas[i][J]; mas[i][J] = z;

}

}

return mas;

}

void Outputf(FILE \*NEW, int m, int n, float \*\*mas)

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < m; j++)

{

printf("%8.2f", mas[i][j]);

fprintf(NEW, "%8.2f", mas[i][j]);

}

printf("\n");

fprintf(NEW, "\n");

}

}

# ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ

## Данная программа предназначена для перемещения наибольшего элемента матрицы в левый верхний угол, путем перестановки сток и столбцов. При запуске программы появляется запрос на ввод необходимых данных. После введения необходимых данных программа выполняет специальные математические вычисления и логические преобразования, в следствие которых выводится ответ.

**ИНСТРУКЦИЯ ПРОГРАММИСТА**

Данная программа предназначена для перемещения наибольшего элемента матрицы в левый верхний угол, путем перестановки сток и столбцов.

Структуры данных, используемых в программе, приведены в таблице 1.

Таблица 1 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| Input, Output | FILE | Имена файлов с входными и выходными данными соответственно |
| n, m | int | Количество строк и столбцов в матрице соответственно |
| mas | float | Обрабатываемая матрица |

Программа разбита на *Х* подпрограмм:

1) float\*\* getmem(float\*\* arr, int n, int m) – подпрограмма для выделения памяти под массив. Структуры данных, используемых в подпрограмме приведена в таблице 2.

Таблица 2 – Структуры данных, используемые в подпрограмме getmem

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| *формальные параметры* | | |
| n, m | int | Количество строк и столбцов в матрице соответственно |
| arr | float\*\* | Обрабатываемая матрица |

**ДЕМОНСТРАЦИОННЫЙ ПРИМЕР**

Результат работы программы для перемещения наибольшего элемента матрицы в левый верхний угол, путем перестановки сток и столбцов, изображен на рисунках 4,5.

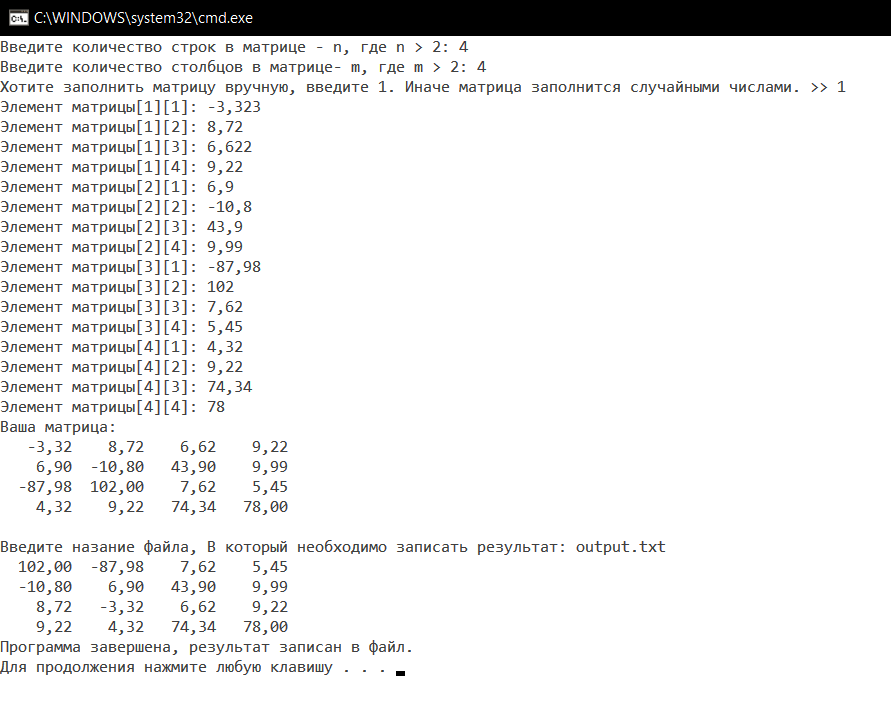


Рисунок 4 – Пример работы программы

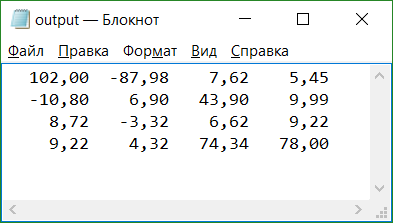


Рисунок 5 – Итоговый файл

Проверим результат работы программы аналитически. Исходя из введенных данных, можно сделать вывод, что максимальный элемент равен 102, который находится в матрице в 3 строке, 2 столбце. Следовательно, по задания мы должны поменять первую строку с третий и после второй столбец с первым:

)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| −87,98 | 102,00 | 7,62 | 5,45 | 102,00 | −87,98 | 7,62 | 5,45 |
| ( 6,90 −10,80 43,90 9,99 ) => −10,80 6,90 43,90 9,99  ( | | | | | | | |
| −3,32 | 8,72 | 6,62 | 9,22 | 8,72 | −3,32 | 6,62 | 9,22 |
| 4,32 | 9,22 | 74,34 | 78,00 | 9,22 | 4,32 | 74,34 | 78,00 |

Если сравнить полученную матрицу и матрицу из файла, можно убедиться, что результат, полученный с помощью программы, верный.

**ВЫВОДЫ**

Создание массива с фиксированным размером подразумевает, что под него выделяется определенная память, соответствующая заданному размеру. Однако это не всегда удобно, а в некоторых случаях бывает необходимо, чтобы количество элементов и соответственно размер выделяемой памяти для массива определялись динамически в зависимости от некоторых условий. В этом случае для создания массива мы можем использовать динамическое выделение памяти.