**ЛАБОРАТОРНАЯ РАБОТА №6**

**«ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ С ПОМОЩЬЮ ФУНКЦИЙ»**

**Цель работы**

Изучить основные принципы обработки двумерных массивов, исследовать способы передачи параметров в функции.

**Задания**

1. Написать программу в соответствии с заданием из методических материалов (вариант 12):

Определить количество строк матрицы, содержащих хотя бы один нулевой элемент. Характеристикой столбца матрицы назовем сумму модулей его отрицательных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с убыванием характеристик.

1. Разработать структурные схемы алгоритма для всех функций решения задачи;
2. Разработать тестовые примеры и выполнить тестирование и отладку написанной программы;

**Текст программы**

Решим задачу для матрицы целочисленного типа данных.

#include<stdio.h>

#include<stdlib.h>

**int** i, j, k; //глобальные параметры

//функция для ввода матрицы m\*n

**void** fill(**int** m, **int** n, **int** \*\*a)

{

**for** (i=0;i<m;i++)

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

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

}

//функция для подсчёта строк матрицы, содержащих 0

**int** zero(**int** m, **int** n, **int** \*\*a)

{

**int** x=0; //количество строк с нулём

**for** (i=0;i<m;i++) //перебор строк и проверка

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

**if** (a[i][j]==0)

{

x++;

**break**;

}

**return**(x);

}

//функция сортировки столбцов матрицы

**void** sort(**int** m, **int** n, **int** \*\*a)

{

**int** sum, //характеристика столбца

har[n], //массив характеристик матрицы

temp; //переменная для обмена

**for** (i=0;i<n;i++) //заполнение har[n]

{

sum=0;

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

**if** (a[j][i]<0)

sum+=abs(a[j][i]);

har[i]=sum;

}

//сортировка столбцов матрицы по убыванию характеристик

**for** (i=1;i<n;i++)

**for** (j=i-1;(har[j]<har[j+1]) && (j>=0);j--)

{

**for** (k=0;k<m;k++)

{

temp=a[k][j];

a[k][j]=a[k][j+1];

a[k][j+1]=temp;

}

temp=har[j];

har[j]=har[j+1];

har[j+1]=temp;

}

}

//функция для вывода матрицы m\*n

**void** output(**int** m, **int** n, **int** \*\*a)

{

printf("\nMatrix:\n");

**for** (i=0;i<m;i++)

{

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

printf("%d ",a[i][j]);

printf("\n");

}

}

main()

{

**int** m, n;

printf("Number of lines -> "); //ввод количества строк

scanf("%d",&m);

printf("Number of columns -> "); //ввод количества столбцов

scanf("%d",&n);

**int** \*\*a = **new** **int** \*[m]; //выделение памяти под массив

**for** (i=0;i<m;i++)

a[i] = **new int** [n];

fill(m, n, a); //функция заполнения массива

//вывод числа строк, содержащих 0

printf("\nNumber of lines with 0 - %d\n",zero(m, n, a));

sort(m, n, a); //сортировка по характеристике

output(m, n, a); //вывод массива

**for** (i=0;i<m;i++) //освобождение памяти

**delete**[]a[i];

**delete**[]a;

}

**Структурная схема программы**

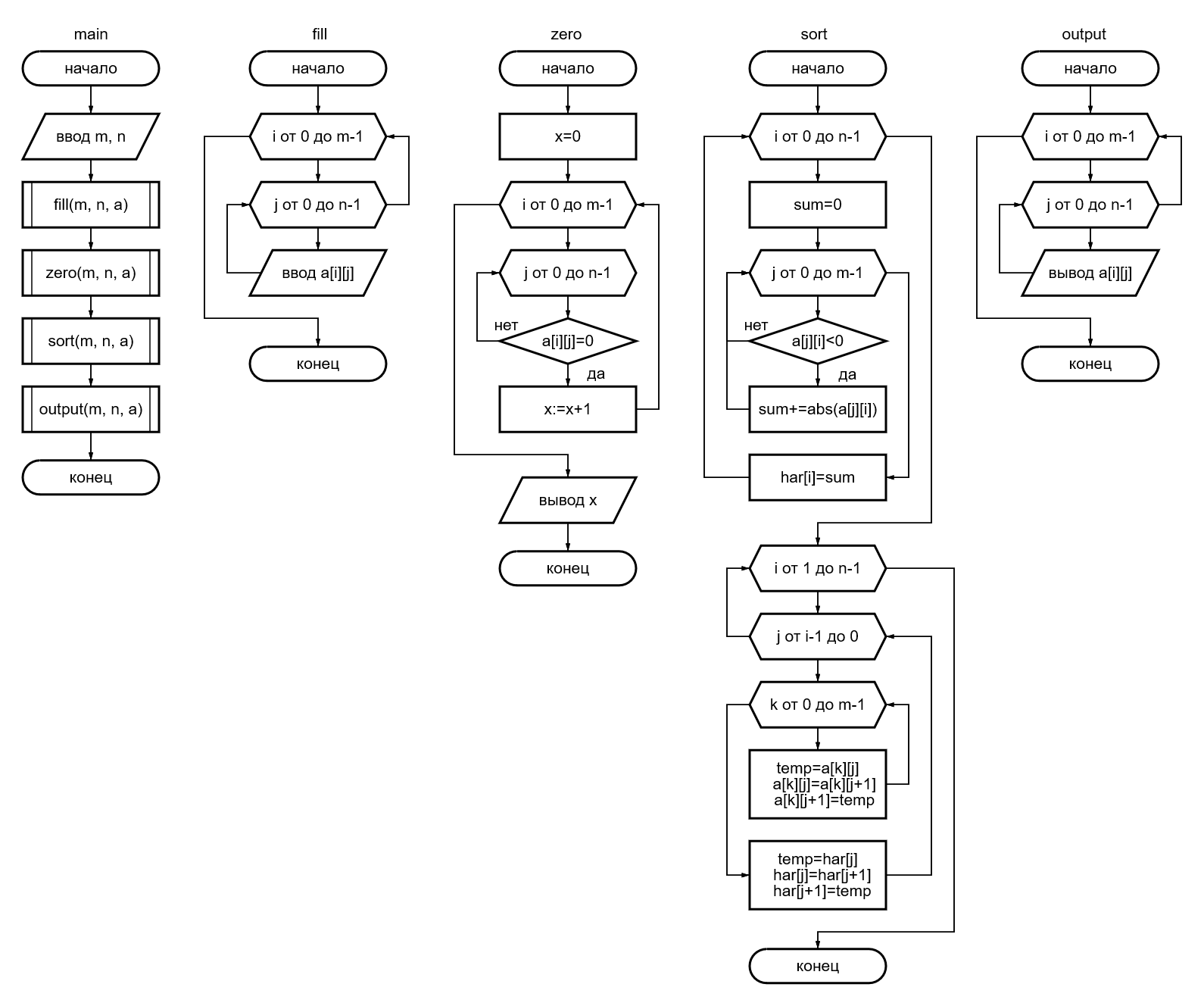


Рисунок 1 – Структурная схема функций программы

**Тестовые примеры**

1. Пусть m=3, n=5. Массив: 1, 5, 3, -1, 0, 0, 9, 4, -1, -2, 1, 8, -3, -1, -2.

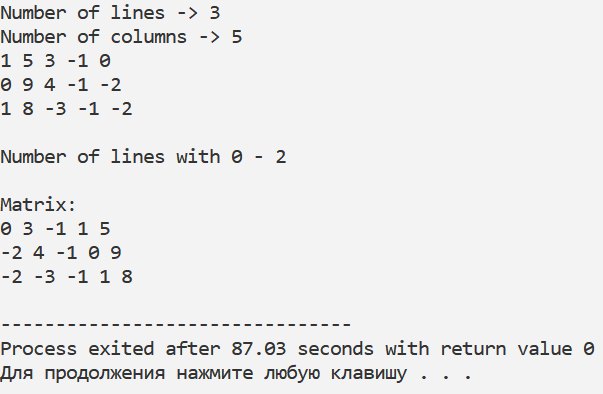


Рисунок 2 – Тестирование программы

1. Пусть m=2, n=1. Массив: 2, 5.

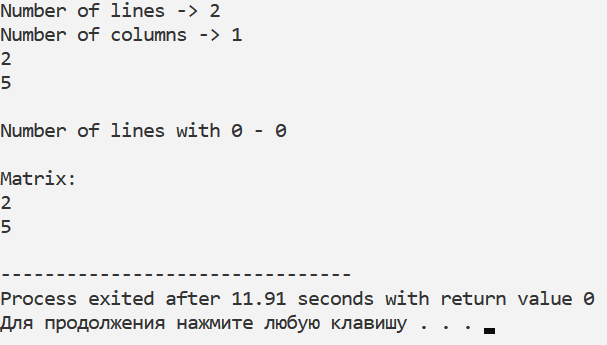


Рисунок 3 – Тестирование программы

1. Пусть m=1, n=5. Массив: 1, -1, -8, 0, -3.

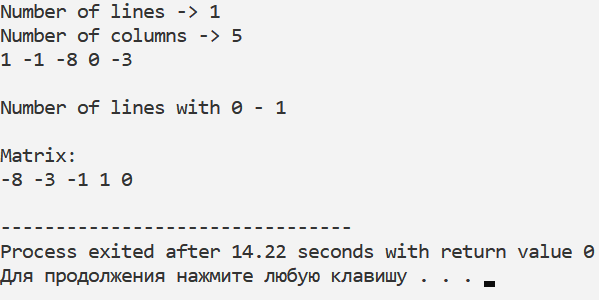


Рисунок 4 – Тестирование программы

1. Пусть m=3, n=4. Массив: 32700, -30000, 0, 32600, -4, 0, 32000, 30500, -29000, 32767, -32767, 0.

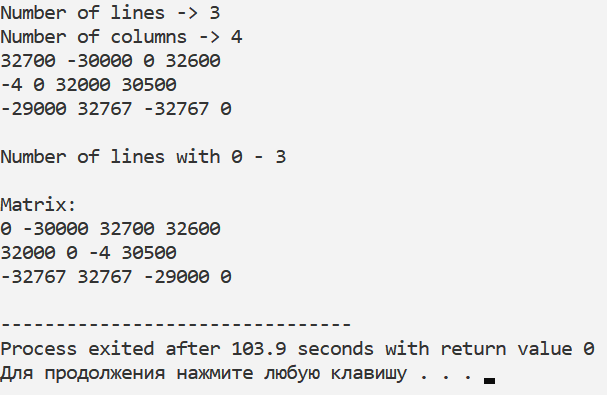


Рисунок 5 – Тестирование программы

Программа вывела верные значения для всех указанных тестовых примеров, на основании чего можно сделать вывод, что она справляется с указанной задачей.

**Вывод**

В ходе работы были изучены особенности работы со статическими и динамическими двумерными массивами в языках C/C++. Освоены навыки простейшей обработки и сортировки двумерных массивов с помощью функций.

Результатом работы стала программа, содержащая функции, которые выполняют ввод, анализ, сортировку и вывод двумерного динамического массива в соответствии с заданными условиями.