# ПОСТАНОВКА ЗАДАЧИ

* Разработать блок-схему и программу, в которой ввести двумерный массив Х, содержащий N строк и M столбцов элементов заданного типа;
* Согласно с вариантом задания № 7, определить количество нулевых элементов в каждой строке матрицы. Исходный массив и результаты вывести на экран, при этом двумерный массив выводить в форме матрицы.
* Разработать тестовые примеры.
* Отладить программу.
* Проанализировать полученные результаты.

# СТРУКТУРНЫЕ СХЕМЫ АЛГОРИТМОВ



Рисунок 1 – Схема алгоритма основной программы



Рисунок 2 – Схема алгоритма функции заполнения матрицы



Рисунок 3 – Схема алгоритма функции печати матрицы на экран



Рисунок 4 – Схема алгоритма функции печати массива на экран



Рисунок 5 – Схема алгоритма функции поиска нулей в строках матрицы и сохранение данных в массив

# ТЕКСТ ПРОГРАММЫ

# #include <stdio.h>

# #include <conio.h>

# #include <stdlib.h>

# int \*\*fillArr(int \*\*arr, size\_t nRow, size\_t nCol){

# for (size\_t i = 0; i < nRow; ++i)

# for(size\_t j = 0; j < nCol; ++j){

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

# }

# return arr;}

# void printMatrix (int \*\*arr, size\_t nRow, size\_t nCol){

# for (size\_t i = 0; i < nRow; ++i){

# for(size\_t j = 0; j < nCol; ++j){

# printf("%+.3d ", arr[i][j]);

# }puts("\n");}}

# int \*searchZero(int \*\*arr, size\_t nRow, size\_t nCol){

# int \*zeroArr = (int\*) calloc(nRow,sizeof(int));

# for (size\_t i = 0; i < nRow; ++i){

# size\_t zeroCount = 0;

# for(size\_t j = 0; j < nCol; ++j){

# if (arr[i][j] == 0) {

# ++zeroCount;

# } }

# zeroArr[i] = zeroCount; }

# return zeroArr;}

# void printArr(int \*arr,size\_t size){

# for(size\_t j = 0; j < size; ++j){ printf("%+.3d ", arr[j]);}}

# int main(){

# system("cls");

# size\_t nRow = 0, nCol = 0;

# printf("Enter size of matrix: \n");

# printf("Enter Row count: ");

# scanf("%d", &nRow);

# printf("Enter Col couunt: ");

# scanf("%d", &nCol);

# int \*\*arr = (int\*\*) calloc(nRow, sizeof(int));

# for(size\_t i = 0; i < nCol; ++i){

# arr[i] = (int\*) calloc(nCol, sizeof(int));

# }

# fillArr(arr, nRow, nCol);

# printMatrix(arr, nRow, nCol);

# int \*zeroArr;

# zeroArr = searchZero(arr,nRow,nCol);

# for(size\_t i = 0; i < nRow; ++i){

# printf("%6d %8d \n", (i+1), zeroArr[i]);

# }

# for(size\_t i = 0; i < nRow; ++i){

# free(arr[i]);

# }

# free(arr);

# free(zeroArr);

# getch();

# return 0;}

# ТЕСТИРОВАНИЕ ПРОГРАММЫ

Для проверки работоспособности программы было разработано несколько тестовых примеров:

* Размерность матрицы выбрана 3х4 и она не содержит нулевых элементов:

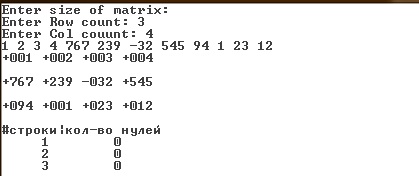


Рисунок 6 – Результат выполнения программы для матрицы, не содержащей нулевых элементов

* Размерность матрицы выбрана 3х3 и она является нулевой:

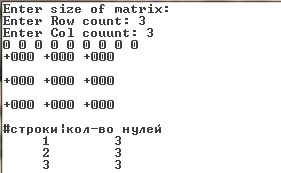


Рисунок 7 – Результат выполнения программы для матрицы содержащей только нулевые элементы

* Размерность матрицы выбрана 4х4 и в каждой строке содержится разное количество нулевых элементов:

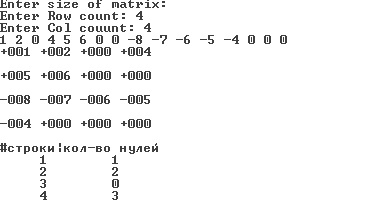


Рисунок 8 – Результат выполнения программы для матрицы содержащей разное количество нулевых элементов в строках

ВЫВОД

При выполнении лабораторной работы была написана программа на языке C, которая позволяет создать двумерный массив размерность которого выбирает пользователь, заполнить его целочисленными значениями введенными с клавиатуры и вывести на экран в виде матрицы, для корректного отображения который у всех чисел предусмотрен вывод знака на экран.

Программа имеет простую структуру, так как имеет всего 5 функций, включая функцию main().

По результатам тестирования, можно сделать вывод, что программа работоспособна и поставленная задача достигнута.