**Министерство образования и наук РФ**

**Федеральное государственное образовательное учреждение**

**Высшего образования**

**Рязанский государственный радиотехнический университет**

**Кафедра ЭВМ**

Отчет по лабораторной работе №6:

“Двумерные массивы”

Вариант 12

Выполнила: ст. гр 640

Тограева К. О.

Проверила:

Доц. к.т.н. Елесина С.И.

Рязань 2017

**Цель работы:**  
 получение навыков работы с динамическими двумерными массивами.

**Вариант №12:**Матрица A имеет седловую точку Aij, если Aij является минимальным элементом в i-й строке и максимальным в j-М столбце.

Дана целочисленная прямоугольная матрица. Определить:

1) количество отрицательных элементов в тех строках, которые содержат хотя бы один нулевой элемент;

2) номера строк и столбцов всех седловых точек матрицы.

**Программирование**

**#include <stdlib.h>  
#include <stdio.h>  
#include <time.h>  
#include <string.h>  
  
#include "main.h"  
// вариант 12  
//Матрица A  
//имеет седловую точку ijA, если ij  
//A является минимальным элементом в  
//i - й строке и максимальным в j  
//-м столбце.  
//Дана целочисленная прямоугольная матрица. Определить:  
//1) количество отрицательных элементов в тех строках, которые  
//содержат хотя бы один нулевой элемент;  
//2) номера строк и столбцов всех седловых точек матрицы.  
  
int isZeroInString(int pInt[n]);  
  
int countElementsLowerZero(int pInt[n]);  
  
int string2int(char stringNumber[]);  
  
char\* itoa(int i, char b[]);  
  
int main() {  
 // для обеспечения рандома  
 srand(time(NULL));  
 int staticArray[n][m] = {0};  
// staticArray = {}; *todo constanta* printf("Массив: \n");  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; ++j) {  
 // заполнить от -4 до 5  
 staticArray[i][j] = rand() % 10 + -5;  
 printf("%i ", staticArray[i][j]);  
 }  
 printf("\n");  
 }  
 findZeroElementIfIsZeroInString(staticArray);  
  
 // 2 task  
 printf("%s", findSedlPoint(staticArray));  
  
 return 0;  
}  
  
void findZeroElementIfIsZeroInString(int staticArray[n][m]) {// 1 task  
 // пробегаемся по сторока  
 for (int i = 0; i < n; ++i) {  
 if (isZeroInString(staticArray[i])) {  
 printf("в строке %i - %i элементов меньше нуля\n"  
 "\n", i, countElementsLowerZero(staticArray[i]));  
 }  
 }  
}  
  
// *todo разнести по функциям*char \*findSedlPoint(int pInt[n][m]) {  
 // массив для хранения точек для вычисления седловой  
 int addArray[n][m] = {0};  
  
 // ищем минимальный элемент  
 for (int i = 0; i < n; i++) {  
 // объявляем, что нулевой элемент в каждой строке - минимальный  
 int min = pInt[i][0];  
 for (int j = 0; j < m; ++j) {  
 if (pInt[i][j] < min) {  
 min = pInt[i][j];  
 }  
 }  
 // в новом массиве заменяем все минимальные значения на двоичную 1  
 for (int j = 0; j < m; ++j) {  
 if (pInt[i][j] == min) {  
 addArray[i][j] = 1;  
 }  
 }  
 }  
  
 // ищем максимальный элемент  
 for (int i = 0; i < m; i++) {  
 // объявляем, что нулевой элемент в каждом столбце - максимальный  
 int max = pInt[0][i];  
 for (int j = 0; j < n; ++j) {  
 if (pInt[j][i] > max) {  
 max = pInt[j][i];  
 }  
 }  
 // если элемент является максимальным, складываем  
 for (int j = 0; j < n; ++j) {  
 if (pInt[j][i] == max) {  
 addArray[j][i] |= 2;  
 }  
 }  
 }  
  
 auto result = (char \*) (calloc(50, sizeof(char)));  
 // выводим седловые элементы, если значение в столбце массива == 3  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < m; j++) {  
 if (addArray[i][j] == 3) {  
 char iStr[12];  
 char jStr[12];  
 char element[12];  
 sprintf(iStr, "%d", i);  
 sprintf(jStr, "%d", i);  
 result = strcat(result, (char \*)("Седловая точка в строчке "));  
 result = strcat(result, itoa(i, iStr));  
 result = strcat(result, (char \*) (" на элементе "));  
 result = strcat(result, itoa(pInt[i][j], element));  
 result = strcat(result, "\n");  
 result = strcat(result, (char \*)("С индексами "));  
 result = strcat(result, itoa(i, iStr));  
 result = strcat(result, " ");  
 result = strcat(result, itoa(j, jStr));  
 result = strcat(result, "\n");  
 }  
 }  
 }  
  
 return result;  
}  
  
int countElementsLowerZero(int pInt[3]) {  
 int elements = 0;  
 for (int i = 0; i < m; ++i) {  
 if (pInt[i] < 0) {  
 elements++;  
 }  
 }  
 return elements;  
}  
  
int isZeroInString(int pInt[3]) {  
 int founded = 0;  
 int i = 0;  
 while (!founded && i < m) {  
 if (pInt[i] == 0) {  
 founded = 1;  
 }  
 i++;  
 }  
 return founded;  
}  
  
char\* itoa(int i, char b[]){  
 char const digit[] = "0123456789";  
 char\* p = b;  
 if(i<0){  
 \*p++ = '-';  
 i \*= -1;  
 }  
 int shifter = i;  
 do{ //Move to where representation ends  
 ++p;  
 shifter = shifter/10;  
 }while(shifter);  
 \*p = '\0';  
 do{ //Move back, inserting digits as u go  
 \*--p = digit[i%10];  
 i = i/10;  
 }while(i);  
 return b;  
}**

Тестовый пример:

Пример 1

Массив:

3 -3 3

4 1 2

-3 -4 -2

-4 -3 -4

3 1 -1

Седловая точка в строчке 1 на элементе 1

С индексами 1 1

Пример 2

Массив:

0 -4 2

-3 1 3

2 -1 -1

2 -4 -2

-5 0 1

в строке 0 - 1 элементов меньше нуля

в строке 4 - 1 элементов меньше нуля

Пример 3

-5 4 2

-3 -1 4

-1 2 0

3 4 3

-3 1 2

в строке 2 - 1 элементов меньше нуля

Седловая точка в строчке 3 на элементе 3

С индексами 3 0

Вывод:

Мы научились работать с двумерными массивами