Министерство образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

ОТЧЕТ

По лабораторной работе №8

ОБРАБОТКА ДВУХМЕРНЫХ МАССИВОВ

С ПОМОЩЬЮ ПРОЦЕДУР И ФУНКЦИЙ

Выполнил: ст. гр. ИТб-11

Куркчи А. Э.

Проверил:

Забаштанский А.К.

Севастополь

2014

1. Цель работы

Изучить основные принципы обработки двухмерных массивов, получить навыки разработки программ блочной структуры, исследовать способы передачи параметров в процедуры и функции.

2. Постановка задач

Вариант №2

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовём сумму её положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

3. Структурная схема

На рисунке 1 изображена схема алгоритма работы программы.

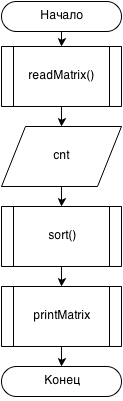


Рисунок 1 – структурная схема

На рисунке 2 изображена схема алгоритма работы процедуры swap.

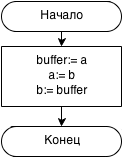


Рисунок 2 – структурная схема процедуры swap

На рисунке 3 изображена схема алгоритма работы процедуры readMatrix.

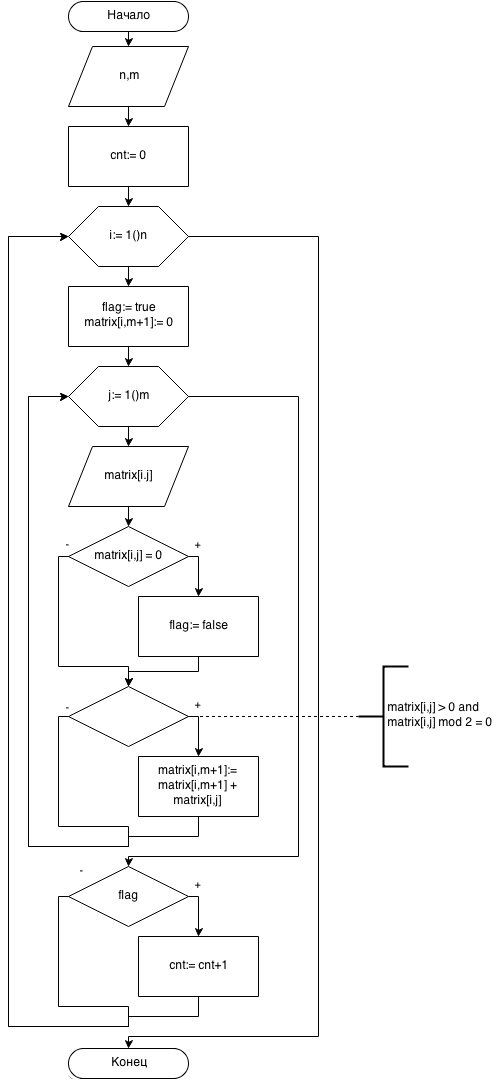


Рисунок 3 – структурная схема процедуры readMatrix

На рисунке 4 изображена схема алгоритма работы процедуры printMatrix.

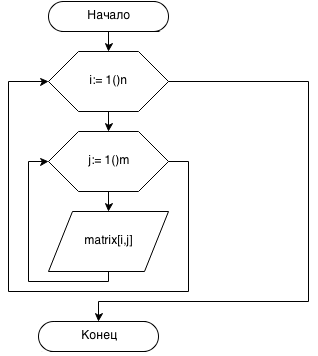
****

Рисунок 4 – структурная схема процедуры printMatrix

На рисунке 5 изображена схема алгоритма работы процедуры printMatrix.

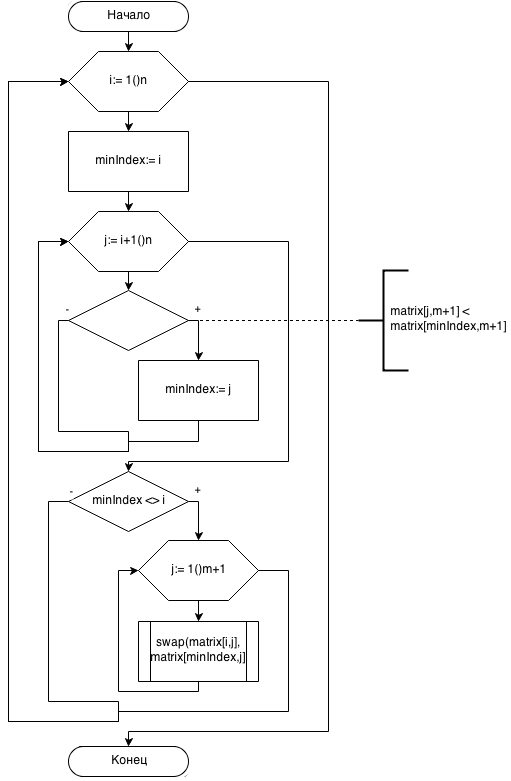


Рисунок 5 – структурная схема процедуры sort

4. Текст программы

**program** lab8;

**type** Tmatrix = **array** [1..1000,1..1000] **of** integer;

**var** n,m,cnt:integer;

matrix:Tmatrix;

**procedure** readMatrix(); {Чтение матрицы}

**var** i,j:integer;

flag:boolean;

**begin**

readln(n,m);

cnt:= 0;

**for** i:= 1 **to** n **do begin**

flag:= true;

matrix[i,m+1]:= 0;

**for** j:= 1 **to** m **do begin**

read(matrix[i,j]);

**if**(matrix[i,j] = 0) **then** flag:= false;

**if**((matrix[i,j] > 0) **and** (matrix[i,j] **mod** 2 = 0)) **then**

inc(matrix[i,m+1],matrix[i,j]);

**end**;

**if**(flag) **then** inc(cnt);

**end**;

**end**;

**procedure** swap(**var** a,b:integer); {Обмен переменных значениями}

**var** buffer:integer;

**begin**

buffer:= a;

a:= b;

b:= buffer;

**end**;

**procedure** sort(); {Сортировка матрици по характеристикам}

**var** i,j,minIndex:integer;

**begin**

**for** i:= 1 **to** n-1 **do begin**

minIndex:= i;

**for** j:= i+1 **to** n **do begin**

**if**(matrix[j,m+1] < matrix[minIndex,m+1]) **then**

minIndex:= j;

**end**;

**if**(minIndex <> i) **then begin**

**for** j:= 1 **to** m+1 **do**

swap(matrix[i,j],matrix[minIndex,j]);

**end**;

**end**;

**end**;

**procedure** printMatrix(); {Печать элементов матрицы}

**var** i,j:integer;

**begin**

**for** i:= 1 **to** n **do begin**

**for** j:= 1 **to** m **do**

write(matrix[i,j],' ');

writeln;

**end**;

**end**;

**begin**

readMatrix();

writeln('Количество строк без нулей: ',cnt);

sort();

writeln('Отсортированная матрица');

printMatrix();

**end**.

5. Тестовые примеры

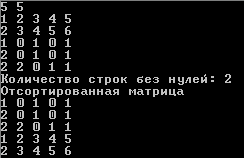


Рисунок 6 – пример №1

На рисунке 6 изображена работа программы с матрицей 5x5 элементов, строки 1 и 2 не содержат нулей, в то время, как строки 3,4 и 5 содержат минимум один ноль. Характеристики строк с 1ой по 5ю: 6, 12, 0, 2 и 4 соответственно.

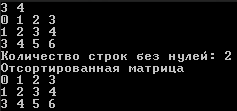


Рисунок 7 – пример №2

На рисунке 7 изображена работа программы с матрицей 3x4 элементов, строки 2 и 3 не содержат нулей, в то время, как 1ая строка содержит ноль в первом элементе. Характеристики строк с 1ой по 3ю: 2, 6 и 10 соответственно.

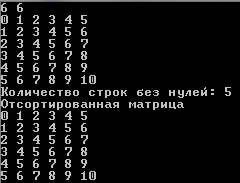


Рисунок 8 – пример №3

На рисунке 7 изображена работа программы с матрицей 3x4 элементов, строки со 2ой по 6ю включительно не содержат нулей, в то время, как 1ая строка содержит ноль в первом элементе. Характеристики строк с 1ой по 6ю: 6, 12, 12, 18, 18 и 24 соответственно.

Вывод

В ходе лабораторной работы были изучены принципы обработки двухмерных массивов, получены навыки разработки программ блочной структуры, исследованы способы передачи параметров в процедуры и функции. Также были рассмотрены методы сортировки матриц.