**Министерство образования и наук РФ**

**Федеральное государственное образовательное учреждение**

**Высшего образования**

**Рязанский государственный радиотехнический университет**

**Кафедра ЭВМ**

Отчет по лабораторной работе №4:

“Двумерные массивы и подпрограммы”

Вариант 11

Выполнила: ст. гр 640

Тограева К. О.

Проверила:

Доц. к.т.н. Елесина С.И.

Рязань 2017

**Цель работы:**  
 получение навыков обработки двумерных массивов

**Вариант №11:** 1) Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2,2), следующий по величине – в позиции (3,3) и т. д., заполнив, таким образом, всю главную диагональ (оформить в виде процедуры).

2) Найти номер первой из строк, не содержащих ни одного положительного элемента (оформить в виде функции).

**Программирование**

**Program**

**MatrixPower;**

**Const**

**// константа,обозначающая количество строк и столбцов в квадратной матрице**

**n : integer = 4;**

**Type**

**// тип - матрица двумерная**

**t = 0..n-1;**

**TMiniArray = array[t] of real;**

**TArray = array [t] of TMiniArray;**

**Var**

**// основная матрица**

**myMatrix : TArray;**

**res : integer;**

**(\***

**Функция, проверяющая, не является ли элемент элементом главной диагонали.**

**Если является, то мы его не учитываем при перестановке, т.к. он уже переставлен**

**\*)**

**function elemGlavnDiagMensheZadannogo(indexForReplacement : integer**

**; i : integer; j : integer) : boolean;**

**begin**

**elemGlavnDiagMensheZadannogo := not((i < indexForReplacement) and (i = j));**

**//если условия не выполняются,то мы там не ищем**

**end;**

**(\***

**Функция, выполняющая поиск элемента по заданному условию**

**\*)**

**function foundWithUpBoard(upBoard : real; var myMatrix : TArray**

**; indexForReplacement : integer) : real;**

**var**

**i, j : integer;**

**max : real;**

**xj, xi : integer; // запомненные максимальные индексы**

**begin**

**max := -maxReal;**

**for i := 0 to n - 1 do**

**for j := 0 to n - 1 do**

**begin**

**if(elemGlavnDiagMensheZadannogo(indexForReplacement, i, j)**

**and (myMatrix[i, j] <= upBoard)**

**and (myMatrix[i, j] >= max)) then**

**begin**

**xi := i;**

**xj := j;**

**max := myMatrix[i, j];**

**end;**

**end;**

**swap(myMatrix[xi, xj], myMatrix[indexForReplacement, indexForReplacement]);**

**foundWithUpBoard := max;**

**end;**

**(\***

**Процедура, перемещающая наибольшие элементы на главную диагональ**

**\*)**

**procedure replaceElements(var myMatrix : TArray);**

**var**

**upBoard : real;**

**indexForReplacement : integer;**

**begin**

**upBoard := maxReal;**

**for indexForReplacement := 0 to n - 1 do**

**upBoard := foundWithUpBoard(upBoard, myMatrix, indexForReplacement);**

**end;**

**(\***

**Процедура,выводящая матрицу на печать**

**\*)**

**procedure writeMatrix(m : TArray);**

**var**

**i, j : integer;**

**begin**

**for i := 0 to n - 1 do**

**begin**

**for j := 0 to n - 1 do**

**write(m[i, j], ' ');**

**writeln();**

**end;**

**writeln();**

**end;**

**(\***

**процедура, заполняющая матрицу случайными значениями в заданном диапазоне**

**\*)**

**procedure fillMatrix(var myMatrix : Tarray);**

**var**

**i, j : integer;**

**begin**

**for i := 0 to n - 1 do**

**for j := 0 to n - 1 do**

**myMatrix[i, j] := random(-10, 1);**

**end;**

**function readByRaws(arr : TMiniArray) : boolean;**

**var**

**i : integer;**

**begin**

**// пока не найдено**

**readByRaws := true;// на самом деле нет ни одного положительного числа**

**for i := 0 to n - 1 do**

**begin**

**// do somethingWItharray**

**if (arr[i] > 0) then**

**readByRaws := false;**

**end;**

**end;**

**(\***

**Процедура,выводящая номер строки и саму строку**

**, если она не содержит ни одного положительного элемента**

**\*)**

**function writeIfNotPlus(myMatrix : TArray) : integer;**

**var**

**i, j : integer;**

**isNegative : boolean;**

**counter : integer;**

**begin**

**counter := -1;**

**while((i < n) and (counter = -1)) do**

**begin**

**readByRaws(myMatrix[i]);**

**if (readByRaws(myMatrix[i])) then**

**counter := i;**

**inc(i);**

**end;**

**writeIfNotPlus := counter;**

**end;**

**begin**

**writeln('Исходная матрица');**

**fillMatrix(myMatrix);**

**writeMatrix(myMatrix);**

**writeln('Матрица с переставленными элементами');**

**replaceElements(myMatrix);**

**writeMatrix(myMatrix);**

**writeln('Выводим строку, не содержащую ни одного положительного элемента');**

**res := writeIfNotPlus(myMatrix);**

**if (res <> -1) then writeln('индекс строки = ', res)**

**else**

**writeln('Нет строки только с отрицательными элементами');**

**end.**

**Тестовый пример:**

**Пример 1**

0 -1 0 -1

0 0 1 -1

0 1 0 -1

0 -1 0 0

Матрица с переставленными элементами

1 -1 0 -1

0 1 0 -1

0 0 0 -1

0 -1 0 0

Выводим строку, не содержащую ни одного положительного элемента

Нет строки только с отрицательными элементами

**Пример 2**

Исходная матрица

-1 -8 0 -5

-9 -6 -3 -5

-5 -10 -2 0

-7 -6 -7 -2

Матрица с переставленными элементами

0 -8 -6 -5

-9 0 -3 -5

-5 -10 -1 -2

-7 -6 -7 -2

Выводим строку, не содержащую ни одного положительного элемента

индекс строки = 2

**Вывод:**

Мы научились работать с двумерными массивами, а также работать с процедурами и функциями