**Національний технічний університет України**

**«Київський політехнічний інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №8**

*Підсумкова робота*

*Виконав студент І курсу*

*Групи ІО-44*

*Барабаш Тарас*

**КИЇВ, 2014**

**ЛАБОРАТОРНА РОБОТА №8**

*Підсумкова робота*

**І. Завдання**

1. Отримати елементи квадратної матриці *А=*⎜*a i,j* ⎜*, i,j=1, ..., n, a i,j = f(i,j),* де *f(i,j)* :

*а i,j = f(i,j),*где *f(i,j) = (|3.7 –j| - 2) (i-4.3-j) + 8,7i.*

Здійснити вивід компонент отриманої матриці.

1. З матриці A отримати компоненти вектора X={xi}, i=1,2,...,n за правилом:

у матриці знайти перший по порядку рядок з найбільшою сумою її елементів і в якості компонент вектора прийняти впорядковані за збільшенням елементи цього рядка.

Здійснити вивід компонент отриманого вектора.

1. Перетворити вектор відповідно до правила:

i-й елемент вектора рівний (хi + хi-1 ) /2, де .

Здійснити вивід компонент перетвореного вектора.

**II. Код програми**

**program** Lab8;

**const** Vmin = 1;

Vmax = 20;

**type**

index = Vmin..Vmax;

Vector = **array** [index] **of** real;

Matrix = **array** [index] **of** Vector;

**var**

A: Matrix;

n, maxi: integer;

V: Vector;

**procedure** InputOfMatrix(**var** A:matrix; n:integer);

**var**

i, j: integer;

**begin**

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

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

A[i, j] := (abs(3.7 - j) - 2) \* (i - power(4.2, j)) + 8.7 \* i;

**end**;

**procedure** OutputOfMatrix(A:Matrix; n:integer);

**var**

i,j : integer;

**begin**

writeln('Result: ');

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

**begin**

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

write(a[i,j]:3:3,' ');

writeln;

**end**;

**end**;

**procedure** FindMax (A:Matrix; n:integer);

**var**

i, j, k: integer;

t, maxs: real;

**begin**

maxi := 1;

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

**begin**

t := 0;

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

t := t + a[i, j];

**if** (t > maxs) **then**

**begin**

maxs := t;

maxi := i;

**end**;

**end**;

**end**;

**procedure** VectorFormation (A:Matrix; n:integer; maxi:integer; **var** V:Vector);

**var** j,k:integer;

**begin**

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

**begin**

k := k + 1;

V[k] := a[maxi, j];

**end**;

**end**;

**procedure** Sort (**var** V:Vector; n:integer);

**var**

i, j, k: integer;

z: real;

**begin**

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

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

**begin**

**if** V[i] > V[j] **then**

**begin**

z := V[j];

V[j] := V[i];

V[i] := z;

**end**;

**end**;

**end**;

**procedure** VectorTranform (**var** V:vector; n:integer);

**var** i:integer;

**begin**

**for** i := n **downto** 2 **do**

V[i] := (V[i]+V[i-1])/2;

**end**;

**procedure** VectorOutput (V:Vector; n:integer);

**var** i:integer;

**begin**

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

write (V[i]:3:3,' ');

writeln;

**end**;

**begin**

write('Rozmir:');read(n);

InputOfMatrix(A,n);

OutputOfMatrix(A,n);

FindMax(A,n);

VectorFormation(A,n,maxi,V);

Sort(V,n);

writeln;

VectorOutput(V,n);

VectorTranform(V,n);

writeln;

VectorOutput(V,n);

**end**.

**III. Висновок**

Оптимізацією даної програми є використання наборів процедур, для структурування програми, виділення окремих операцій і блоків та, щоб уникнути зайвих змінних.