Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Informatica si Ingineria Sistemelor

**RAPORT**

Lucrarea de laborator nr.5

la Programarea Calculatoarelor

A efectuat:

st. gr. C-171 D. Melniciuc

A verificat:

dr., conf.univ. M. Kulev

Chişinău 2017

**Lucrarea de laborator nr.5**

**Tema:** **Tablouri bidimensionale (matrici)**

**Scopul lucrarii practice:**

Studierea şi utilizarea instrucţiunilor de bază ale limbajului pentru prelucrarea şi modificarea masivelor bidimensionale.

**Notiuni de baza:**

Tabourile în C/C++ pot avea mai multe dimensiuni. O modalitate comună de a le folosi este utilizarea matricelor cu linii şi coloane, fiind vorba în acest caz despre tablouri cu două

dimensiuni (bidimensional). Tablou bidimensional - succesiune de locaţii de memorie

recunoscute prin acelaşi identificator şi prin poziţia fiecăreia în cadrul şirului. Poziţia este dată printr-o suită de două numere pozitive (indexi), care reprezintă cele două dimensiuni (linie şi

coloană).

Sintaxa de declarare a unui tablou bidimensional (matrice) este:

Tip nume[dim1][dim2];

unde:

Tip - poate fi unul din tipurile de bază folosit (int,

float,char,...) sau un tip definit de utilizator (articole, obiecte);

nume – numele prin care va fi referită matricea;

dim1 – numărul de linii din matrice;

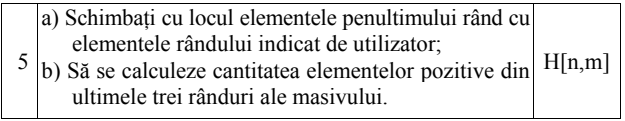
dim2 – numărul de coloane din matrice.

**Condiţia problemei (sarcina de lucru):**

Să se scrie un program ce va prelucra un masiv

bidimensional cu n linii şi m coloane, având următoarele condiţii:

**Varianta 5:**



**Mersul lucrarii:**

[2]Pentru a programa la calculator e necesara urmarirea unui algoritm strict. Pasii sunt urmatorii:

-cunoasterea problemei;

Citirea si intelegerii corecte a problemei.

-analiza datelor;

Studierea datelor problemei pentru a destinge datele de tip input, output si date intermediare.

-elaborarea algoritmului prin schema logica;

Elaborarea schemei logice a programului.

-codul (textului) programei;

Scrierea codului propriu-ziz.

-testarea;

Rularea problemei pentru a determina erori logice sau de sintaxa.

-verificarea rezultatelor

Introducerea diferitor date pentru a verifica mai multe posibilitati.

-analiza rezultatelor si formarea concluziilor;

Formare concluziilor adecvate dupa efectuarea pasilor anteriori.

Analiza datelor:

In problema data am urmatoarele conditii

a) Schimbaţi cu locul elementele penultimului rând cu

elementele rândului indicat de utilizator;

b) Să se calculeze cantitatea elementelor pozitive din

ultimele trei rânduri ale masivului.

**Schema logica:**

“Introduceti numarul de linii:”

n

“Introduceti numarul de coloane:”

m

“Aflam cantitatea elementelor positive”

i=0; i<n; i++

i=n-3; i<n; i++

j=0; j<m; j++

j=0; j<m; j++

h[i+1][j+1]

h[i][j]>0

h[i][j]

k++

i=0; i<n; i++

j=0; j<m; j++

h[i][j]

K =

\n

“Schimbati cu locul elementele penultimului rand”

“Introduceti valori penultimului rand”

i=n-2; i<n-1; i++

j=0; j<m; j++

h[i+1][j+1]

h[i][j]

\n

i=0; i<n; i++

j=0; j<m; j++

h[i][j]

**Codul (textul) programului in limbajul C:**

#include <conio.h>

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

int main()

{

int i, j, n, m;

int h[10][10];

printf("Introduceti numarul de linii: ");

scanf("%d", &n);

printf("Introduceti numarul de coloane: ");

scanf("%d", &m);

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

printf("h[%d][%d] = ", i+1, j+1);

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

}

}

printf("\n\rMatricea\n\r");

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

printf("%3d", h[i][j]);

}

printf("\n\r");

}

printf("\n\rAflam cantitatea elementelor pozitive\n\r");

int k = 0;

for (i=n-3; i<n; i++)

{

for (j=0; j<m; j++)

{

if(h[i][j]>0) k++;

}

}

printf("K = %d\n", k);

printf("\n\r\n\rSchimbati cu locul elementele penultimului rand\n\r");

printf("Introduceti valori penultimului rand\n\r");

for (i=n-2; i<n-1; i++)

{

for (j=0; j<m; j++)

{

printf("h[%i][%d] = ", i, j);

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

}

}

printf("\n\rMatricea cu penultimul rand schimbat\n\r");

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{printf("%3d", h[i][j]);}

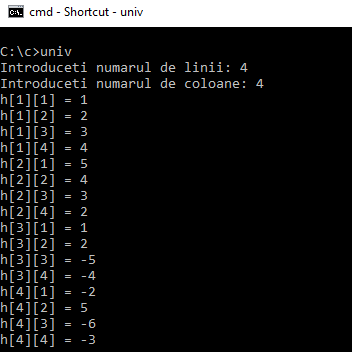
printf("\n\r");

}

return 0;

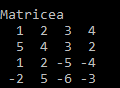
}

**Rezultatele obtinute:**

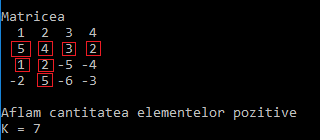
****

Introducem numarul de linii si coloane

Apoi dam valoare numerica pe pozitiile matriciei

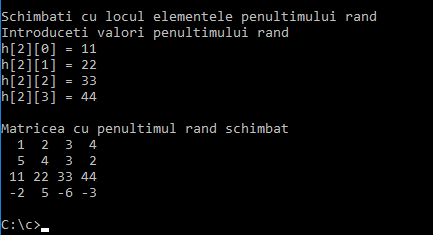
Afisam matricea

Dupa conditie trebuie sa aflam cantitatea elementelor pozitive din

ultimele trei rânduri ale masivului.

Dupa a doua conditie trebuie sa schimbam locul elementele penultimului rând cu

elementele rândului indicat de utilizator



**Bibliografie:**

[1] Derevlenco, Vasile și alț. Programare : Indicaţii de laborator. Partea 1. – Ch.: Tehnica – UTM, 2014. <http://library.utm.md>

[2]Conspectul si notele de la lectiile teoretice

**Concluzie:**

In urma efectuarii programului si a rularii acestuia au fost obtinute anumite abilitati in domeniu programarii si anume studierea tabelelor bidimensionale sau 2D.

**Avantaje**

Putem da orice numarul de linii si coloane.

Deasemenea putem da valoare pentru fiecare pozitie a matriciei.

Programul este interactiv.

**Dezavantje**

Dupa conditie numarul de linii trebuie sa fie cel putin egal cu 3.

**Dezvoltarea programului**

Micsorarea numarului liniilor din cod.

Restructurarea codului pentru un nivel de citire mai usor.