

Ministerul Educației, Culturii și Cercetării
Universitatea Tehnică a Moldovei



Departamentul Ingineria Software și Automatică

RAPORT

Lucrarea de laborator nr. 4
la Programarea Calculatoarelor
Varianta 18

A efectuat:
st. gr. TI-206
A verificat:
Lector universitar

Cătălin Pleșu
Vitalie Mititelu

Chișinău - 2020

Lucrarea de laborator nr. 4

Tema : Prelucrarea tablourilor bidimensionale (matricelor) în limbajul C

Scopul : Studierea posibilităților și mijloacelor limbajului C pentru programarea algoritmilor de prelucrare a tablourilor bidimensionale.

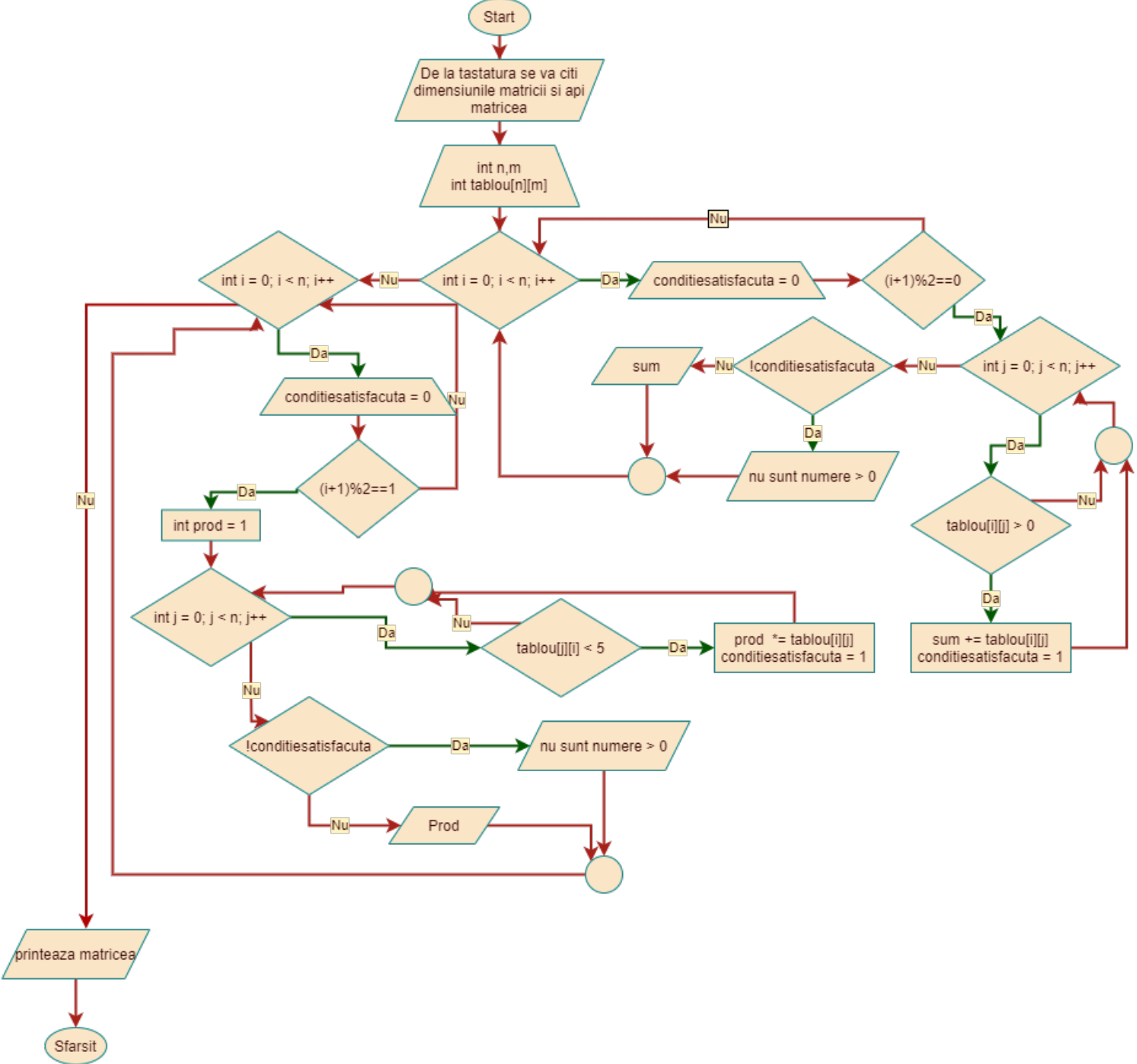
Sarcina: Scrieți un program care citește de la tastatură numărul $n > 1$ de rânduri și numărul $m > 1$ de coloane ale tabloului bidimensional (matricei), apoi citește de la tastatură aceste $n \times m$ elemente ale tabloului, efectuează calculele indicate în variantă și afișează pe ecran rezultatul:

Varianta 18. Să se determine suma elementelor pozitive din liniile pare și produsul elementelor mai mici ca 5 din coloanele impare.

Rezumat succint la temă :

- Acest program va opera cu o matrice bidimensională care va fi citită de la tastatură.
- Când vor fi afișate pozițiile elementelor matricii și numărul liniei sau coloanei, la acest număr voi adăuga 1 pentru a fi mai ușor de înțeles pentru persoanele care nu au cunoștințe de programare.
- Folosesc două for-uri pentru a calcula suma elementelor pozitive din liniile pare și tot două foruri pentru a calcula produsul elementelor mai mici ca 5 din coloanele impare.
- În programul dat pot apărea probleme dacă pe linii sau coloane apar elemente care nu satisfac condițiile astfel că am adăugat o variabilă care cu scopul logic care să urmărească dacă condiția a fost activată cel puțin o singură dată.
- Programul ne va afișa rândul sau coloana și rezultatul operației efectuate.
- Programul dat înainte de sfârșit printează matricea inițială pentru comoditatea mea.

Schema logică a algoritmului :



Cod sursă în limbajul C :

```
#include <stdio.h>
```

```
int main()
{
    int n, m;
    printf("\n\n\t\t\t");
    printf("Lucrarea de laborator nr. 4 Varianta 18\n\n\n");
    printf("Acest program va citi o matrice si va calcula :\nsuma elementelor pozitive din liniile pare si\
nprodusul elementelor mai mici ca 5 din coloanele impare\n");
    printf("nr de randuri = ");
    scanf("%d", &n);
    printf("nr de coloane = ");
    scanf("%d", &m);
    int tablou[n][m];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
        {
            printf("elementul [%d %d] = ", i + 1, j + 1);
            scanf("%d", &tablou[i][j]);
        }

    //varianta 18
    //cerinte
    //1 suma elementelor pozitive din liniile pare
    //2 produsul elementelor < 5 din coloanele impare

    for (int i = 0; i < n; i++)
    {
        if ((i + 1) % 2 == 0)
        { //operatiile pentru rand par
            int conditiesatisfacuta = 0;
            printf("\nrand par %d :\n", i + 1);
            int sum = 0;
            for (int j = 0; j < m; j++)
            {
                if (tablou[i][j] > 0)
                {
                    sum += tablou[i][j];
                    conditiesatisfacuta = 1;
                }
                printf("%d ", tablou[i][j]);
            }
            if (!conditiesatisfacuta)
                printf("\nnici un numare > 0");
            else
                printf("\nsum = %d", sum);
        }
        printf("\n");
    }

    for (int i = 0; i < m; i++)
    {
        int conditiesatisfacuta = 0;
        if ((i + 1) % 2 == 1)
        { //operatiile pentru coloana impara
            printf("\ncoloana impara %d :", i + 1);
```

```

int prod = 1;
for (int j = 0; j < n; j++)
{
    if (tablou[j][i] < 5)
    {
        prod *= tablou[j][i];
        conditiesatisfacuta = 1;
    }
    printf("\n%d", tablou[j][i]);
}
if (!conditiesatisfacuta)

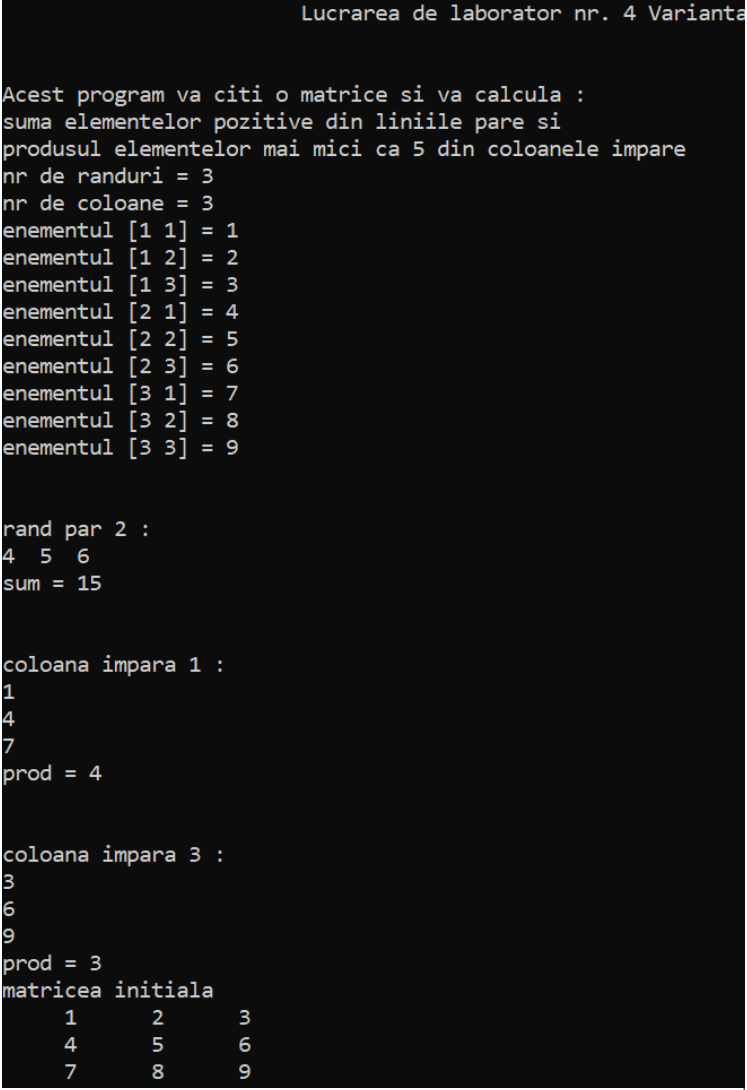
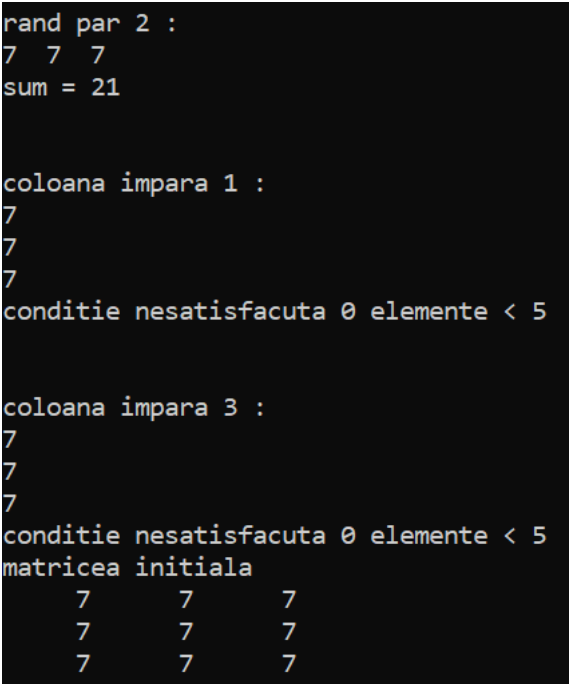
    printf("\nconditie nesatisfacuta 0 elemente < 5");

else
    printf("\nprod = %d", prod);
}
printf("\n");
}

//printeaza tabloul / matricea
printf("matricea initiala");
for (int i = 0; i < n; i++)
{
    printf("\n");
    for (int j = 0; j < m; j++)
    {
        printf("%6d ", tablou[i][j]);
    }
}
return 0;
}

```

Verificarea datelor de ieșire :

Date de intrare	Date de ieșire	Screenshot
nr de randuri = 3 nr de coloane = 3 enementul [1 1] = 1 enementul [1 2] = 2 enementul [1 3] = 3 enementul [2 1] = 4 enementul [2 2] = 5 enementul [2 3] = 6 enementul [3 1] = 7 enementul [3 2] = 8 enementul [3 3] = 9	rand par 2 : 4 5 6 sum = 15 coloana impara 1 : 1 4 7 prod = 4 (doar 1 și 4 < 5) coloana impara 3 : 3 6 9 prod = 3	 <pre>Lucrarea de laborator nr. 4 Varianta 1 Acest program va citi o matrice si va calcula : suma elementelor pozitive din liniile pare si produsul elementelor mai mici ca 5 din coloanele impare nr de randuri = 3 nr de coloane = 3 enementul [1 1] = 1 enementul [1 2] = 2 enementul [1 3] = 3 enementul [2 1] = 4 enementul [2 2] = 5 enementul [2 3] = 6 enementul [3 1] = 7 enementul [3 2] = 8 enementul [3 3] = 9 rand par 2 : 4 5 6 sum = 15 coloana impara 1 : 1 4 7 prod = 4 coloana impara 3 : 3 6 9 prod = 3 matricea initiala 1 2 3 4 5 6 7 8 9</pre>
nr de randuri = 3 nr de coloane = 3 enementul [1 1] = 7 enementul [1 2] = 7 enementul [1 3] = 7 enementul [2 1] = 7 enementul [2 2] = 7 enementul [2 3] = 7 enementul [3 1] = 7 enementul [3 2] = 7 enementul [3 3] = 7	rand par 2 : 7 7 7 sum = 21 coloana impara 1 : 7 7 7 conditie nesatisfacuta 0 elemente < 5 coloana impara 3 : 7 7 7 conditie nesatisfacuta 0 elemente < 5 matricea initiala 7 7 7 7 7 7 7 7 7	 <pre>rand par 2 : 7 7 7 sum = 21 coloana impara 1 : 7 7 7 conditie nesatisfacuta 0 elemente < 5 coloana impara 3 : 7 7 7 conditie nesatisfacuta 0 elemente < 5 matricea initiala 7 7 7 7 7 7 7 7 7</pre>

nr de randuri = 3 nr de coloane = 3 elementul [1 1] = -2 elementul [1 2] = -2 elementul [1 3] = -1 elementul [2 1] = -3 elementul [2 2] = -4 elementul [2 3] = -5 elementul [3 1] = -6 elementul [3 2] = -7 elementul [3 3] = -8	rand par 2 : -3 -4 -5 nici un numare > 0 coloana impara 1 : -2 -3 -6 prod = -36 coloana impara 3 : -1 -5 -8 prod = -40 matricea initiala -2 -2 -1 -3 -4 -5 -6 -7 -8	<pre> rand par 2 : -3 -4 -5 nici un numare > 0 coloana impara 1 : -2 -3 -6 prod = -36 coloana impara 3 : -1 -5 -8 prod = -40 matricea initiala -2 -2 -1 -3 -4 -5 -6 -7 -8 </pre>
--	---	---

Analiza datelor de ieșire :

1. Dacă elementele de pe rândurile pare sunt mai mari ca 0 și dacă elementele de pe coloanele impare sunt mai mici ca 5 pe ecran vor fi afișate sumele și produsurile respective.
2. Dacă toate elementele de pe coloanele impare sunt mai mari ca 5 atunci se va afișa că nu a fost nici un număr mai mic ca cinci.
3. Dacă toate elementele de pe rândurile pare sunt negative atunci se va afișa că nici un număr nu este mai mare ca zero.
4. În cazul în care pe coloane există numere mai mari ca 5, se va calcula doar produsul elementelor mai mici ca 5, dacă avem un singur element el se înmulțește cu 1 și se afișează produsul.
5. În cazul când pe rând există elemente negative acestea sunt ignorate în obținerea sumei.

Concluzii :

Limbajul C oferă mijloace pentru prelucrarea tablourilor bidimensionale. Sunt capabil să utilizez limbajul C pentru a prelucra tablourile bidimensionale. Pe viitor dacă nu voi fi mai atent la condițiile date voi avea probleme. Schema logică a programului face codul ușor de citit însă într-un program mai mare ar fi cam complicată de urmărit. Programele în limbajul C sunt destul de plictisitoare fiind că nu au interfață grafică, puțini utilizatori ar fi dispuși să le utilizeze. Codul pe care l-am scris poate fi optimizat prin eliminarea a unui set de cicluri condițiile cărora pot fi integrate în ciclul precedent deși acest fapt ar împiedica printarea rândului și coloanei separat. Datele de ieșire sunt calculate conform condiției.