

Ministerul Educației, Culturii și Cercetării
Universitatea Tehnică a Moldovei



Departamentul Ingineria Software și Automatică

RAPORT

Lucrarea de laborator nr. 3
la STRUCTURI DE DATE ȘI ALGORITMI

Tema : Algoritmi de prelucrare a fișierelor text
Varianta 18

A efectuat:
st. gr. TI-206

Cătălin Pleșu

A verificat:
Lector universitar

Mititelu Vitalie

Chișinău – 2021

Cuprins

Sarcina pentru varianta 18:.....	3
Rezumat succint la temă:.....	4
Cod sursă in limbajul C :	5
main.c.....	5
fisiere.c.....	6
matrici.c.....	9
Screenshot-uri cu programul în acțiune:	11
Date de intrare	11
Date de ieșire.....	11
Verificare datelor de ieșire	11
Fișierele text pentru exemplele precedente	12
Introducerea manuală a amtricilor în fișier:.....	13
Concluzii :	14

Tema: Algoritmi de prelucrare a fișierelor text

Scopul: Programarea algoritmilor de prelucrare a fișierelor text prin utilizarea funcțiilor, pointerilor, alocării dinamice a memoriei în limbajul C.

Sarcina pentru varianta 18:

1. Să se scrie programul care creează un fișier text în care se memorează două matrice
 - a. pe prima linie numărul de linii și numărul de coloane ale primei matrice, separate printr-un spațiu;
 - b. pe fiecare din liniile următoare, în ordine, elementele fiecărei linii din matrice, separate prin câte un spațiu;
 - c. în continuare a doua matrice, în aceeași formă.
2. Să se scrie programul care înmulțește matricele din acest fișier.
3. Rezultatul se va memora în alt fișier, în aceeași formă.
4. Dacă înmulțirea nu e posibilă, se va scrie ca rezultat un mesaj de eroare.
5. Matricele sunt suficient de mici pentru a putea fi încărcate în memorie ($n, m \leq 5$).

Rezumat succint la temă:

- Pentru a opera cu fișiere avem nevoie de biblioteca **stdlib.h** iar pentru a citi de la tastatură și pentru a afișa la ecran folosesc librăria **stdio.h**.
- Am decis să împart programul respective în mai multe fișiere pentru a naviga mai ușor codul, astfel programul dat este constituit din 3 fișiere .c împărțite după funcționalitate.
- Pe lângă **fopen** și **fclose** am utilizat două funcții pentru a opera cu fișiere: **fprintf** și **fscanf**, aceste funcții permit citirea și scrierea unui text formatat, exact ce am nevoie pentru a satisface condițiile lucrării (separarea numerelor prin spații).
- Am utilizat funcțiile legate de manipularea fișierelor text în fișierul **fișiere.c**.
- În fișierul **matrici.c** am implementat funcțiile pentru **alocarea dinamică** a memoriei pentru o matrice și eliberării ei, de asemenea am scris funcțiile pentru citirea și printarea matricii. În ultimul rând înmulțirea matricilor.
- Merită de menționat că pentru a satisface condiția 4 a fost nevoie să mă abțin de la a forța ca indiferent de dorința utilizatorului numărul de rânduri a celei de a doua matrici să fie egalat cu numărul de coloane a primei matrici.

Cod sursă în limbajul C :

main.c

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

#include "hdr/fisiere.c"
#include "hdr/matrici.c"

int main()
{
    int n1, n2, m1, m2;
    int **M1 = NULL, **M2 = NULL;
    printf("Program la SDA, lab 3 varianta 18\n");
    printf("Citirea a 2 matrici dintr-
un fisier si scrierea produsului lor in alt fisier\n");
    int optiune = 123;
    while (optiune)
    {
        printf("(1) Citirea matricilor de la tastatura\n");
        printf("(2) Importarea matricilor din fisier\n");
        printf("(0) pentru a iesi din program\n");
        scanf("%d", &optiune);
        switch (optiune)
        {
            case 1:
                printf("matricea 1\n");
                M1 = citireaMatrice(&n1, &m1);
                printf("matricea 2\n");
                printf("nr de randuri trebuie sa fie egal cu %d!!!\n", m1);
                M2 = citireaMatrice(&n2, &m2);
                scriereaInFisier(n1, n2, m1, m2, M1, M2);
                break;
            case 2:
                dimensiunileMatricilor(&n1, &m1, &n2, &m2);
                M1 = creareMatrice(n1, m1);
                M2 = creareMatrice(n2, m2);
                citireaMatricilor(M1, M2);
                break;
            default:
```

```

        break;
    }
    printf("matricea 1\n");
    printeazaMatrice(n1, m1, M1);
    printf("matricea 2\n");
    printeazaMatrice(n2, m2, M2);
    int **M = NULL;
    if (m1 == n2)
    {
        M = multiplicareaMatricilor(n1, n2, m2, M1, M2);
        printf("produsul matricilor\n");
        printeazaMatrice(n1, m2, M);
    }
    else
        printf("nu poate fi calculat produsul\n");

    scriereProdus(n1, n2, m1, m2, M);
    freeMatrice(M, n1);
    freeMatrice(M1, n1);
    freeMatrice(M2, n2);
}

return 0;
}

```

fisiere.c

```

const char fisier1[20] = "lab3_in.txt";
const char fisier2[20] = "lab3_out.txt";
void scriereaInFisier(int n1, int n2, int m1, int m2, int **M1, int **M2)
{
    FILE *fisier = NULL;
    fisier = fopen(fisier1, "w");
    if (!fisier)
    {
        printf("nu s-a putut deschde acest fisier\n");
        return;
    }
    fprintf(fisier, "%d %d\n", n1, m1);
    for (int i = 0; i < n1; i++)
    {

```

```

        for (int j = 0; j < m1; j++)
            fprintf(fisier, "%d ", M1[i][j]);

        fprintf(fisier, "\n");
    }
    fprintf(fisier, "%d %d\n", n2, m2);
    for (int i = 0; i < n2; i++)
    {
        for (int j = 0; j < m2; j++)
            fprintf(fisier, "%d ", M2[i][j]);

        fprintf(fisier, "\n");
    }

    fclose(fisier);
}

void dimensiunileMatricilor(int *n1, int *m1, int *n2, int *m2)
{
    FILE *fisier = NULL;
    fisier = fopen(fisier1, "r");
    if (!fisier)
    {
        printf("in memorie nu exista nici un fisier\n");
        return;
    }
    fscanf(fisier, "%d %d", n1, m1);
    int M1[*n1][*m1];
    for (int i = 0; i < *n1; i++)
        for (int j = 0; j < *m1; j++)
            fscanf(fisier, "%d ", &M1[i][j]);

    fscanf(fisier, "%d %d", n2, m2);
    int M2[*n2][*m2];
    for (int i = 0; i < *n2; i++)
        for (int j = 0; j < *m2; j++)
            fscanf(fisier, "%d ", &M2[i][j]);
    fclose(fisier);
}

```

```

void citireaMatricilor(int **M1, int **M2)
{
    int n, m;
    FILE *fisier = NULL;
    fisier = fopen(fisier1, "r");
    if (!fisier)
    {
        printf("nu exista acest fisier\n");
        return;
    }

    fscanf(fisier, "%d %d", &n, &m);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            fscanf(fisier, "%d ", &M1[i][j]);

    fscanf(fisier, "%d %d", &n, &m);
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            fscanf(fisier, "%d ", &M2[i][j]);
    fclose(fisier);
}

void scriereProdus(int n, int a, int b, int m, int **M)
{
    FILE *fisier = NULL;
    fisier = fopen(fisier2, "w");
    if (!fisier)
    {
        printf("nu s-a putut deschde acest fisier\n");
        return;
    }

    if (a != b)
    {
        fprintf(fisier, "Aceste matrici nu pot fi inmultite");
        fclose(fisier);
        return;
    }

    fprintf(fisier, "%d %d\n", n, m);
    for (int i = 0; i < n; i++)

```



```

{
    for (int j = 0; j < m; j++)
        fprintf(fisier, "%d ", M[i][j]);

    fprintf(fisier, "\n");
}

fclose(fisier);
}

```

matrici.c

```

void printeazaMatrice(int n, int m, int **matrice)
{
    printf("%d %d\n", n, m);
    printf("\n");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
            printf("%d ", matrice[i][j]);

        printf("\n");
    }
    printf("\n");
}

int **createMatrice(int n, int m)
{
    int **matrice = NULL;
    matrice = (int **)calloc(n, sizeof(int *));
    for (int i = 0; i < n; i++)
        matrice[i] = (int *)calloc(m, sizeof(int));
    return matrice;
}

int **multiplicareaMatricilor(int n, int m, int p, int **M1, int **M2)
{
    int **M = createMatrice(n, p);
    for (int i = 0; i < n; i++)
        for (int k = 0; k < p; k++)
            for (int j = 0; j < m; j++)
                M[i][k] += M1[i][j] * M2[j][k];
}

```

```

    return M;
}
int **citireaMatrice(int *n, int *m)
{
    printf("dati nr. de randuri ");
    scanf("%d", n);
    printf("dati nr. de coloane ");
    scanf("%d", m);
    int **M = creareMatrice(*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", &M[i][j]);
        }
    return M;
}
void freeMatrice(int **matrice, int n)
{
    for (int i = 0; i < n; i++)
        free(matrice[i]);
    free(matrice);
    matrice = NULL;
}

```

Screenshot-uri cu programul în acțiune:

Meniul principal

Program la SDA, lab 3 varianta 18
 Citirea a 2 matrici dintr-un fisier si scrierea produsului lor in alt fisier
 (1) Citirea matricilor de la tastatura
 (2) Importarea matricilor din fisier
 (0) pentru a iesi din program

Date de intrare	Date de ieșire	Verificare datelor de ieșire												
matricea 1 dari nr. de randuri 2 dari nr. de coloane 2 elementul [1][1] = 1 elementul [1][2] = 2 elementul [2][1] = 3 elementul [2][2] = 4 matricea 2 nr de randuri trebuie dari nr. de randuri 2 dari nr. de coloane 3 elementul [1][1] = 1 elementul [1][2] = 2 elementul [1][3] = 3 elementul [2][1] = 4 elementul [2][2] = 5 elementul [2][3] = 6	matricea 1 2 2 1 2 3 4 matricea 2 2 3 1 2 3 4 5 6 produsul matricilor 2 3 9 12 15 19 26 33	<table><tr><td></td><td>C₁</td><td>C₂</td><td>C₃</td></tr><tr><td>1</td><td>9</td><td>12</td><td>15</td></tr><tr><td>2</td><td>19</td><td>26</td><td>33</td></tr></table> E		C ₁	C ₂	C ₃	1	9	12	15	2	19	26	33
	C ₁	C ₂	C ₃											
1	9	12	15											
2	19	26	33											
matricea 1 dari nr. de randuri 1 dari nr. de coloane 1 elementul [1][1] = 1 matricea 2 nr de randuri trebuie dari nr. de randuri 2 dari nr. de coloane 2 elementul [1][1] = 1 elementul [1][2] = 1 elementul [2][1] = 1 elementul [2][2] = 1	matricea 1 1 1 1 matricea 2 2 2 1 1 1 1 nu poate fi calculat produsul													

matricea 1

dari nr. de randuri 3

dari nr. de coloane 3

elementul [1][1] = 1

elementul [1][2] = 2

elementul [1][3] = 3

elementul [2][1] = 4

elementul [2][2] = 5

elementul [2][3] = 6

elementul [3][1] = 7

elementul [3][2] = 8

elementul [3][3] = 9

matricea 2

nr de randuri trebuie

dari nr. de randuri 3

dari nr. de coloane 4

elementul [1][1] = 1

elementul [1][2] = 2

elementul [1][3] = 3

elementul [1][4] = 4

elementul [2][1] = 5

elementul [2][2] = 6

elementul [2][3] = 7

elementul [2][4] = 8

elementul [3][1] = 9

elementul [3][2] = 10

elementul [3][3] = 11

elementul [3][4] = 12

matricea 1

3 3

1 2 3

4 5 6

7 8 9

matricea 2

3 4

1 2 3 4

5 6 7 8

9 10 11 12

produsul matricilor

3 4

38 44 50 56

83 98 113 128

128 152 176 200

	C ₁	C ₂	C ₃	C ₄
1	38	44	50	56
2	83	98	113	128
3	128	152	176	200

Site-ul utilizat pentru verificarea inmulțirii matricilor:

<https://matrix.resish.com/multCalculation.php>

Fișierele text pentru exemplele precedente

lab3_in.txt

1	2	2	
2	1	2	
3	3	4	
4	2	3	
5	1	2	3
6	4	5	6

lab3_out.txt

```
lab3_in.txt lab3_out.txt
```

1	2	3
2	9	12 15
3	19	26 33

1	1	1
2	1	
3	2	2
4	1	1
5	1	1

1 Aceste matrici nu pot fi inmult

lab3_in.txt	lab3_out.txt
1 3 3	1 3 4
2 1 2 3	2 38 44 50 56
3 4 5 6	3 83 98 113 128
4 7 8 9	4 128 152 176 200
5 3 4	
6 1 2 3 4	
7 5 6 7 8	
8 9 10 11 12	

Introducerea manuală a amtricilor în fișier:

```

Program la SDA, lab 3 varianta 18
Citirea a 2 matrici dintr-un fisier si scrierea produsului lor in alt fisier
(1) Citirea matricilor de la tastatura
(2) Importarea matricilor din fisier
(0) pentru a iesi din program
2
matricea 1
3 3

1 2 3
1 2 3
1 2 3

matricea 2
3 10

1 2 3 4 5 1 2 3 4 5
1 2 3 4 5 1 2 3 4 5
1 2 3 4 5 1 2 3 4 5

produsul matricilor
3 10

6 12 18 24 30 6 12 18 24 30
6 12 18 24 30 6 12 18 24 30
6 12 18 24 30 6 12 18 24 30

```

lab3_in.txt	lab3_out.txt
1 3 3	1 3 10
2 1 2 3	2 6 12 18 24 30 6 12 18 24 30
3 1 2 3	3 6 12 18 24 30 6 12 18 24 30
4 1 2 3	4 6 12 18 24 30 6 12 18 24 30
5 3 10	
6 1 2 3 4 5 1 2 3 4 5	
7 1 2 3 4 5 1 2 3 4 5	
8 1 2 3 4 5 1 2 3 4 5	

Concluzii :

- Deși acest laborator a fost relativ unul simplu datorită efectuării lui am aflat cum pot scrie numere într-un fișier text sau mai bine zis am lămurit confuzia legată de faptul că în fișierele binare nu am putut vedea numere (speranța mea de a vedea numere în fișierul binar a fost destul de stupidă 😅).
- Operarea cu fișiere este esențială pentru toate programele, ba chiar mai mult, în timpul scrierii programului citirea datelor dintr-un fișier poate salva mult timp, de aceea probabil voi utiliza fișiere chiar și la laboratoarele care nu cer acest lucru dacă va fi cazul.
- Un fișier cu cod de câteva sute de linii este destul de greu de navigat și de înțeles de aceea consider că este bine de separat programul în mai multe fișiere.
- În acest caz particular este mult mai eficient să introducem matricile în fișierul text decât să le citim de la tastatură.