

LABORATOR NR. 8

Operații de intrare ieșire cu fișiere

Problemele rezolvate până acum cereau citirea datelor de intrare de la tastatură și afișarea rezultatelor pe ecranul monitorului. Dar, putem prelucra informații care se găsesc pe disc sau putem scrie rezultatele unor prelucrări pe disc (sau un alt suport de memorie externă care permite citirea și / sau scrierea unor informații).

Pe disc informațiile sunt organizate în fișiere. Fiecare fișier de pe disc are un marker de sfârșit de fișier care poate fi identificat într-un program **C** folosind aceeași modalitate ca și atunci când se apasă pe tastatură combinația de taste CTRL/Z (pentru Windows) sau CTRL/D (pentru Linux).

Pentru a putea lucra cu fișiere mai întâi trebuie declarată o variabilă care va fi asociată fișierului cu care lucrăm. Această asociere se face la deschiderea fișierului.

Declarația variabilei care va fi asociată fișierului are forma generală:

```
FILE * numeVariabila;
```

unde **numeVariabila** este numele variabilei. Vom avea astfel de declarații pentru fiecare fișier cu care vrem să lucrăm.

Deoarece pot apărea erori la deschiderea unui fișier, trebuie verificat dacă, după deschiderea fișierului, variabila **numeVariabila** are valoarea egală cu zero. În acest caz fișierul nu s-a putut deschide și programul trebuie încheiat.

Secvența de cod pentru deschiderea unui fișier pentru citire (vrem să citim din el) în cazul în care am avut declarația:

```
FILE *in;
```

este

```
in = fopen("intrare.txt","r");
if (in == 0)
{
    fprintf(stderr, "Eroare la deschiderea fișierului intrare.txt.\n");
    exit(EXIT_FAILURE);
}
```

unde **intrare.txt** este numele fișierului din care citim datele.

Funcția **exit**, ca și constanta **EXIT_FAILURE** sunt declarate în fișierul header **stdlib.h**, ca urmare și acest fișier trebuie inclus în program.

Secvența de cod pentru deschiderea unui fișier pentru scriere (vrem să scriem din el) în cazul în care am avut declarația:

```
FILE *out;
```

este

```
out = fopen("iesire.txt", "w");
if (out == 0)
{
    fprintf(stderr, "Eroare la deschiderea fișierului iesire.txt.\n");
    exit(EXIT_FAILURE);
}
```

unde **iesire.txt** este numele fișierului în care scriem rezultatele.

După ce am terminat lucrul cu un anumit fișier trebuie să-l închidem. Închiderea fișierului se face folosind funcția **fclose**, astfel:

```
fclose(in);      /* Inhide fisierul pentru citire */
fclose(out);     /* Inhide fisierul pentru scriere */
```

Citirea dintr-un fișier se face prin folosirea funcției **fscanf**, iar scrierea într-un fișier se face folosind funcția **fprintf**.

Cele două funcții sunt asemănătoare funcțiilor **scanf** și **printf**, doar că înaintea specificării formatului de citire sau de scriere, trebuie să specificăm fișierul din care citim sau în care scriem.

Exemplu 1

- a) Vrem să citim din fișierul care are asociată variabila **in** valoarea variabile **n** care este de tip întreg. Instrucțiunea va fi:

```
fscanf (in, "%d", &n);
```



fișier din
care se
citește

- b) Citirea unui real în dublă precizie pentru care avem declarația
- ```
double a;
```

se va face cu instrucțiunea:

```
fscanf (in , "%lf", &a);
```

fișier din  
care se  
citește

c) Scrierea în fișierul care are asociată variabila **out**:

```
fprintf (out , "Valoarea lui n este %d.\n", n);
```

fișier în  
care se  
scrie

```
fprintf (out , "a = %5.3lf si n = %d\n", a, n);
```

fișier în  
care se  
scrie

Toate regulile de scriere enunțate pentru funcțiile **printf** și **scanf** rămân valabile și pentru **fscanf** și **fprintf**.

Implicit tastatura (dispozitivul standard de intrare) are asociată variabila **stdin**, iar monitorul (dispozitivul standard de ieșire) are asociată variabila **stdout**. Astfel instrucțiunea:

`scanf("%d", &n);` este echivalentă cu `fscanf( stdin , "%d", &n);`

fișier din  
care se  
citește

iar instrucțiunea

`printf("%lf", a);` este echivalentă cu `fprintf ( stdout , "%lf", a);`

fișier în  
care se  
scrie

**Observație:****Pentru Linux**

După lansarea în execuție și executarea unui program care lucrează cu fișiere, afișarea conținutului unui fișier în fereastra **Terminal** se face folosind comanda **cat** sau **less**.

Exemplu:

**cat in.dat**

afișează tot conținutul fișierului **in.dat** pe terminal. Se recomandă pentru fișiere cu un număr mic de linii (cât încap pe un ecran).

**less in.dat**

afișează numai primele linii din fișierul **in.dat** trecerea la liniile următoare se face prin tastarea **Enter** sau săgeată în jos. Ieșirea din fișier se face prin apăsarea tastei **q**.

**Pentru Windows**

După lansarea în execuție și executarea unui program care lucrează cu fișiere, afișarea conținutului unui fișier din fereastra MS-DOS se face folosind comanda **type**.

Exemplu: afișarea conținutului fișierului **in.dat** (dacă presupunem că lucrăm în directorul – folder-ul **Temp**) se face tastând **type in.dat**. Adică pe ecran va fi următoarea linie:

**c:\Temp>type in.dat**

**Pentru rezolvarea corectă a problemelor ce urmează vă rog să citiți cu atenție fiecare enunț de la început până la sfârșit.**

**Rezolvările se fac folosind proiecte și fără variabile globale.**

**TEMA 1****Problema 1.1.**

Să se scrie un program care citește de la tastatură un tablou de maximum 20 de numere întregi, calculează suma valorilor vectorului și scrie rezultatul în fișierul **rezP11.dat**.

Se vor scrie funcții adecvate pentru citirea unui vector de la tastatură, scrierea unui vector pe monitor, calcularea sumei valorilor elementelor unui vector.

**Problema 1.2.**

Într-o stație meteorologică se măsoară, în fiecare zi la ora 7, temperatura la 1,5 m deasupra solului. La anumite intervale de timp, măsurate în zile, se realizează rapoarte care

cuprind: temperatura minimă, temperatura maximă, temperatura medie din intervalul de timp considerat, precum și media geometrică a valorilor absolute ale temperaturilor măsurate în intervalul de timp dat (această valoare va fi folosită apoi în alte prelucrări necesare pentru o analiză meteorologică completă).

Intervalul de timp pentru care se fac prelucrările poate fi de maxim 31 de zile (raportul se face numai pentru zile din aceeași lună).

Să se scrie un program care citește informațiile necesare de la tastatură și generează raportul cerut (ca în exemplul de mai jos) în fișierul **raport.dat**.

Valorile temperaturilor măsurate (se iau în considerare valori întregi pentru temperaturile măsurate) se stochează într-un vector, iar, pentru individualizarea prelucrărilor, în raport se va indica și luna pentru care se fac prelucrările.

*Indicație: Pentru media geometrică se va folosi funcția **pow**, iar pentru determinarea valorii absolute se va folosi funcția **abs**.*

Se vor scrie funcții adecvate pentru citirea unui vector, scrierea unui vector, calcularea valorii maxime a elementelor unui vector, calcularea valorii minime a elementelor unui vector, calcularea mediei aritmetice a valorilor elementelor unui vector, calcularea mediei geometrice a valorilor elementelor unui vector.

**Exemplu de raport care trebuie generat de program** (textul scris cu **albastru** reprezintă mesajele scrise de program, textul scris cu **verde** reprezintă valorile datelor de intrare, iar textul scris cu **roșu** reprezintă valorile rezultatelor calculate de program).

Raport de temperatura pentru ultimele 3 zile din luna **februarie**.

Temperaturile citite sunt: **-7**, **1**, **-5**.

Valoarea maxima a temperaturii este: **1**.

Valoarea minima a temperaturii este: **-7**.

Valoarea medie a temperaturii este: **-3.667**.

Media geometrica a temperaturilor este **3.2711**.

Final raport.

*Atenție:* în acest exemplu culorile sunt folosite doar a evidenția de unde provin informațiile citite. Programul nu va folosi culori diferite pentru scrierea raportului și nici nu va genera un chenar.

**Problema 1.3.**

Să se scrie un program care citește un tablou de maximum 30 de numere întregi dintr-un fișier numit **inP12.dat**, afișează pe monitor vectorul citit, ordonează elementele vectorului crescător prin metoda bulelor și scrie într-un fișier cu numele **rezP12.dat** vectorul inițial și vectorul ordonat crescător. Se vor scrie funcții pentru citirea unui vector de numere întregi dintr-un fișier, scrierea unui vector de numere întregi pe monitor, scrierea unui vector de numere întregi în fișier și ordonare prin metoda bulelor.

**Problema 1.4.**

În Pădurea cu alune au case **n** pitici. Cămara fiecărei case are **m** rafturi, câte un raft pentru fiecare din cele **m** produse pe care piticii le pot cumpăra de la magazinul aflat la marginea pădurii. De la magazin poate cumpăra doar starostele piticilor și numai în prima zi din lună. Pentru a putea face cumpărăturile necesare, la fiecare sfârșit de lună starostele piticilor trece pe la casa fiecărui pitic și ia comanda acestuia, după care se duce la magazin și face cumpărăturile necesare.

Toate produsele din magazin se vând la bucată.

Trebuie să se calculeze:

- 1) Care sunt stocurile de alimente ale piticilor după ce starostele s-a întors de la magazin?
- 2) Cât trebuie să plătească fiecare pitic dacă se cunosc prețurile pe bucată ale celor **m** produse existente în cămările piticilor?
- 3) Câte zile trebuie să muncească fiecare pitic pentru a achita nota de plată la magazin? Se presupune că piticii nu fac alte cheltuieli în afara celor pentru mâncare și că pentru o zi de muncă fiecare pitic primește **G** galbeni. (Se va face rotunjirea la număr întreg de zile, de exemplu: dacă rezultă 2.34 zile, se va afișa atât valoarea exactă calculată – în cazul acesta 2.34 zile – cât și rotunjirea – în acest caz 3 zile -.)

**Date de intrare**

Datele de intrare se citesc din fișierul **PITICI.IN** care are pe prima linie un număr real care reprezintă plata pentru o zi de lucru (**G**).

Pe următoarea linie se găsesc numărul **n** de pitici din pădure și numărul **m** de rafturi din cămara fiecărui pitic.

Pe următoarele **n** linii se găsesc stocurile din cele **m** alimente pe care le are fiecare pitic date ca numere întregi pozitive câte **m** pe o linie separate de câte un spațiu.

Pe următoarele **n** linii se găsesc comenzile pentru cele **m** alimente pe care le face fiecare pitic date ca numere întregi pozitive câte **m** pe o linie separate de câte un spațiu.

Pe ultima linie se găsesc prețurile celor **m** alimente necesare piticilor date ca numere reale separate de câte un spațiu.

### Date de ieșire

Rezultatele obținute trebuie înscrise în fișierul **PITICI.OUT**, iar datele de intrare trebuie afișate pe monitor pentru a verifica citirea lor corectă.

## TEMA 2

### Problema 2.1.

Scrieți un program care folosește două funcții proprii care convertesc textul citit linie cu linie din fișierul **text.in** (citirea se face până se întâlnește sfârșitul de fișier) în litere mari sau litere mici după cum se indică prin dialog de la tastatură. Textul astfel transformat se va înscrie într-un alt fișier al cărui nume se citește de la tastatură.

#### Observații:

- 1) Nu se va face apel la funcțiile de conversie din bibliotecă.
- 2) Aveți grijă atunci când citiți numele fișierului de la tastatură că funcția `fgets` păstrează și caracterul `'\n'` în șir. Acest caracter trebuie eliminat prin înlocuirea lui cu `'\0'`.

### Problema 2.2.

Firma "Trioda" comercializează componente electronice și are două magazine în **Pădurea cu alune**. Cele două magazine se numesc **Trioda1** și **Trioda2** și comercializează aceleași sortimente de componente electronice. La sfârșitul fiecărei luni se face un inventar al stocurilor de componente electronice din cele două magazine pentru a determina valoarea stocurilor de componente electronice și valoarea totală a mărfii. După inventar trebuie generat un raport ca în exemplul următor (convențiile de culoare sunt cele de la problema 1.2):

Raport inventar pentru firma Trioda.

Firma Trioda comercializează 5 tipuri de componente electronice.

Pret / bucata pentru fiecare componenta electronica = (125.3, 3.15, 574, 98.35, 4.25)

Stocuri magazin Trioda1 = (23, 675, 12, 5, 170)

Stocuri magazin Trioda2 = (12, 25, 3, 25, 30)

Total stocuri firma = (35, 700, 15, 30, 200)

Valoare stocuri magazin Trioda1 = (2881.90, 2126.25, 6888.00, 491.75, 722.50)

Valoare stocuri magazin Trioda2 = (1503.60, 78.75, 1722.00, 2458.75, 127.50)

Valoare stocuri firma = (4385.50, 2205.00, 8610.00, 2950.50, 850.00)

Valoarea totală marfă = 19001.00 lei.

Sfarsit raport inventar.

Să se scrie un program care citește informațiile necesare și

- generează un raport de inventar pentru firma **Trioda** după modelul dat;
- calculează valoarea stocurilor firmei și valoarea totală a mărfii după o lună de la inventarul inițial considerând că stocurile au rămas aceleași (ceea ce s-a vândut a fost imediat completat), dar toate prețurile au crescut cu o anumită valoare procentuală, valoare care va fi citită de la tastatură (de exemplu, creșterea prețurilor a fost de 0.7%).

Firma **Trioda** poate avea pe stoc la un moment dat maxim 50 de tipuri de componente electronice.

Se vor scrie funcții adecvate pentru: citirea unui vector, afișarea unui vector (conform modelului dat), înmulțirea a doi vectori, suma a doi vectori, suma valorilor elementelor a doi vectori, înmulțirea unui vector cu un scalar.

Datele de intrare se găsesc în fișierul **trioda.dat**, iar rezultatele (raportul de inventar inițial și noile valori ale stocurilor) vor fi afișate pe ecran

### Problema 2.3.

Pentru o matrice pătratică cu elemente numere reale se definesc următoarele norme:

$$\|A\|_{\infty} = \max_i \sum_j |a_{ij}| \quad (\text{maximul sumelor de pe fiecare linie})$$



$$\|A\|_1 = \max_j \sum_i |a_{ij}| \quad (\text{maximul sumelor de pe fiecare coloană})$$

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2}$$

unde  $|x|$  definește modulul numărului real  $x$ .

Să se scrie un program care folosește ca fișier de intrare fișierul **matIn.dat**.

Acest fișier are pe prima linie două numere întregi reprezentând numărul de linii și numărul de coloane ale unei matrice cu elemente numere reale.

Pe următoare linii sunt elementele matricei date pe linii.

Astfel, dacă fișierul cuprinde liniile

2 3

1.23 3.14 5.2

17.34 32.123 5.19

atunci matricea citită are două linii și trei coloane și este de forma:  $\begin{bmatrix} 1.23 & 3.14 & 5.2 \\ 17.34 & 32.123 & 5.19 \end{bmatrix}$

Să se calculeze normele definite mai sus. Se va realiza un dialog prin care să se aleagă opțiunea de prelucrare. Rezultatele vor fi înscrise pe monitor.

#### Problema 2.4.

Un teren de formă dreptunghiulară este împărțit în parcele, dispuse pe  $n$  rânduri, pe fiecare rând fiind  $m$  parcele. Se măsoară altitudinea fiecărei parcele (se consideră că o parcelă are altitudine constantă). Altitudinile tuturor parcelelor se memorează într-un tablou cu  $n$  linii și  $m$  coloane. Valoarea elementului de pe linia  $i$  și coloana  $j$  a tabloului este altitudinea celei de a  $j$ -a parcele de pe linia  $i$ .

Să se afișeze coordonatele “parceleuror vârfo”. O “parcelă vârfo” are altitudinea strict mai mare decât a tuturor vecinilor săi.

Citirea datelor de intrare se face dintr-un fișier, iar rezultatele se vor înscrie într-un alt fișier.