

Teme și aplicații - Fișiere text

Formatori:

Tutor: [Stângaciu Valentin](#) 

Tutor: [Belu Claudiu-Marcel](#) 

+3

Data de începere a cursului:

 25.09.2023

 [Utilizatori înscriși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) [S1-L-AC-CTIRO1-PC](#) [Laborator 13: Fișiere](#) [Teme și aplicații - Fișiere text](#)

Teme și aplicații - Fișiere text

ATENȚIE ! Pentru rezolvarea acestor probleme este necesar ca programul să citească/scrie fișierele folosind funcțiile de bibliotecă dedicate. NU se rezolvă problemele folosind redirectarea intrării și ieșirii standard !

1. Se consideră un fișier csv (comma separated values) în care pe fiecare linie se află mai multe câmpuri separate prin virgulă. Fișierul conține informații despre filme. Pe fiecare linie se află câte o înregistrare ce reprezintă un film cu date despre acesta, date ce sunt separate prin virgulă. O linie are următoarea configurație (cap de tabel)

```
year,imdb,title,test,clean_test,binary,budget,domgross,intgross,code,budget_2013$,domgross_2013$,intgross_2013$,period  
code,decade code
```

Se va citi acest fișier în mod optim. Pentru fiecare înregistrare se vor extrage următoarele date: titlu, an, buget (title, year, budget). Se va proiecta și implementa o structură de date corespunzătoare. Se vor stoca aceste date într-un tablou alocat dinamic. Se precizează că se consideră necunoscută dimensiunea fișierului (a numărului de linii) - se va alocă minimul de memorie necesar pentru a stoca toate informațiile. Programul va scrie în fișierul out1.txt toate filmele sortate după anul apariției, în fișierul out2.txt toate filmele sortate după titlu iar în fișierul out3.txt toate filmele sortate după buget.

Fișierul de test poate fi descărcat folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/movies.csv
```

Se vor realiza mai multe variante de procesare și extragere de informații dintr-o linie de date: folosind strtok, folosind strchr, algoritmul propriu prin parcurgere caracter cu caracter a unei linii deja citite din fișier.

2. Se citește de la tastatură maxim 100 numere reale, până la întâlnirea numărului 0. Să se sorteze aceste numere și să se scrie într-un fișier, toate numerele fiind pe o singură linie, separate prin | (bară verticală).
3. Se consideră un fișier csv (comma separated values) în care pe fiecare linie se află mai multe câmpuri separate prin semnul punct-virgulă. Fișierul conține informații statistice oferite de Institutul National de Statistica prin platforma OpenData a Guvernului Romaniei (<https://data.gov.ro/>) despre plata indemnizațiilor de creșterea copilului pe o anumită lună din anul 2021. Fișierul are următorul cap de tabel ce descrie câmpurile din liniile conținute

```
Judet;Populatie;Beneficiari plătiți;Beneficiari suspendați la sfârșit de lună;Suma totală plătită drepturi curente;Alte plăți
```

Programul va citi acest fișier folosind funcțiile dedicate pentru citirea fișierelor text, va extrage *câmpurile judet, beneficiari platiti si suma total platita drepturi curente*, și va stoca datele într-un tablou alocat dinamic, de structuri de date corespunzătoare, de dimensiune minimă necesară. Se cere să se implementeze tipuri de date utilizator. Se va implementa de asemenea o funcție de sortare după judet și o funcție de sortare după suma totală platită drepturi curente. Programul va scrie în fișierul rezultat.txt pe câte un rând diferit judetul cu cea mai mare populație, judetul cu cea mai mare sumă din câmpul alte plăți.

Fișierul se poate descărca folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/indemnizatie.csv
```

4. Se consideră un fișier text ce conține un număr necunoscut de linii de text de o dimensiune necunoscută. Să se scrie un program ce citește un astfel de fișier și stochează datele într-un tablou alocat dinamic de string-uri C alocate dinamic. Se va folosi o amprentă minimă de

memorie. Se vor scrie de asemenea și următoarele funcții

- o funcție `void write_sort(char **lines, char *filepath);` - funcția va prelua tabloul de linii, le va sorta alfabetic și le va scrie în fișierul cu calea `filepath`. Se va realiza conex și o funcție de sortare a tabloului de linii `lines`. Se poate folosi `bubble-sort`. Pentru compararea alfabetică a două linii se poate folosi funcția `strcmp(...)`
- o funcție `void write_random(char **lines, char *filepath);` - funcția va prelua tabloul de linii și le va scrie în fișierul referit prin calea `filepath` în ordine aleatoare a liniilor. Hint ! Se poate folosi funcția `rand(...)`;
- o funcție `uint32_t find_word(char **lines, char *word);` - funcția va căuta cuvântul referit prin `word` în tabloul de linii `lines` și va returna numărul de apariții.

Fișierul de test se poate descărca folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/scrisoare.txt
```

5. Se consideră un fișier text ce conține în general text, de o dimensiune necunoscută, programul considerând-o infinită. Să se scrie un program care citește acest fișier și scrie în fișierul `histo.txt` histragama literelor din fișier. Prin histrograma literelor (mari și mici) se înțelege distribuția în procente a acestora, procentul de apariție din total. În fișierul `histo.txt` se vor scrie linii sub forma : litera - numar apariții.
- Exemplu:

```
a - 16 %  
b - 2 %  
c - 17.8 %
```

Fișierul de test se poate descărca folosind următoarea comandă:

```
wget http://staff.cs.upt.ro/~valy/pt/scrisoare.txt
```

◀ Considerații teoretice

Sari la...

Teme și aplicații - Fișiere binare ►

✉ Contactați serviciul de asistență

Sunteți conectat în calitate de 
S1-L-AC-CTIRO1-PC

Meniul meu

Profil

Preferințe

Calendar

 ZOOM

Română (ro)

English (en)

Română (ro)

Rezumatul păstrării datelor

Politici utilizare site