

## Laborator 7 PA - DICȚIONARE, FUNCȚII

### FUNCȚII

1.
  - a) Scrieți o funcție care să citească de la tastatură o listă cu elemente numere întregi. Numărul de elemente ale listei și elementele sale se vor citi în cadrul funcției
  - b) Scrieți o funcție care primește ca parametru o secvență s, un element x și, opțional, doi indici i și j și returnează poziția primului element mai mare decât x din s[i:j] (dacă i sau j nu se specifică, atunci comportamentul va fi cel de la feliere) și -1 în caz că nu există un astfel de element.
  - c) Scrieți un program care, folosind apeluri utile ale funcției definite anterior, afișează mesajul "Da" în cazul în care o listă de numere întregi, citită de la tastatură, este sortată strict descrescător sau mesajul "Nu" în caz contrar. Aceeași cerință și pentru o listă de cuvinte.
2.
  - a) Scrieți o funcție cu număr variabil de parametri care să furnizeze numărul natural obținut prin alipirea cifrelor maxime ale numerelor naturale nenule primite ca parametri. De exemplu, pentru numerele 4251, 73, 8 și 133 funcția trebuie să returneze numărul 5783.
  - b) Scrieți o funcție cu 3 parametri nenuli de tip întreg a,b și c care să verifice dacă aceștia pot fi considerați ca fiind numere scrise în baza 2 sau nu, folosind apeluri utile ale funcției definite anterior. De exemplu, pentru numerele 1001, 11 și 100 funcția trebuie să returneze valoarea True, iar pentru numerele 1001, 17 și 100 trebuie să returneze valoarea False.
3. Scrieți o funcție cu număr variabil de parametri care să caute un cuvânt dat în mai multe fișiere text. Funcția va scrie într-un fișier text câte o linie pentru fiecare fișier text de intrare, astfel: numele fișierului text de intrare și apoi numerele de ordine ale liniilor pe care apare cuvântul dat în acel fișier (numerotate de la 1) sau un mesaj corespunzător dacă fișierul nu conține cuvântul respectiv. Antetul funcției va fi: **cautare\_cuvant(cuv, nume\_fis\_out, \*nume\_fis\_in)**. Se vor număra aparițiile cuvântului fără a face diferența între literă mare și literă mică. De exemplu, prin apelul **cautare\_cuvant("floare","rez.txt", "eminescu.txt", "paunescu.txt")** se va căuta cuvântul "floare" în fișierele text "eminescu.txt" și "paunescu.txt", iar rezultatul căutării va fi scris în fișierul text "rez.txt".

eminescu.txt	paunescu.txt	rez.txt
Floare albastra Inc-o gura - si dispare... Ca un stalp eu stam in luna! Ce frumoasa, ce nebuna E albastra-mi, dulce floare! Si te-ai dus, dulce minune, S-a murit iubirea noastra - Floare-albastra! floare-albastra!... Totusi este trist in lume!	Aceeasi floare Iubito, pe prispa ti-am pus O floare din lumea de sus, O floare din insula Marte, O floare a cailor moarte. O floare a vietii vecine Si care miroase a tine. Iubito, pe prispa tu ai O floare din insula Rai.	eminescu.txt 1 5 8 paunescu.txt 1 3 4 5 6 9

#### 4. (Problemă din modelul de subiect dicționare)

Fișierul text *cinema.in* conține programul dintr-o zi al unui lanț de cinematografe. Fiecare linie din fișier are următoarea structură:

*nume\_cinematograf % nume\_film % ore\_de\_difuzare*

unde *nume\_cinematograf* este un șir de caractere reprezentând numele unui cinematograf, *nume\_film* este numele unui film (numele cinematografului și al filmului sunt formate din cuvinte separate prin câte un spațiu și nu conțin caracterul '%'), iar *ore\_de\_difuzare* este un șir de caractere conținând orele (sub forma hh:mm) la care este programat filmul în cinematograf, orele fiind separate prin câte un spațiu. Un exemplu de astfel de fișier este:

**cinema.in**

```
Cinema 1 % Minionii 2 % 12:30 18:30
Cinema 3 % Elfii cofetari % 10:30 12:30
Cinema 2 % Minionii 2 % 15:00 18:30 20:30
Cinema 1 % Elfii cofetari % 10:00 12:30
Cinema 2 % Gasca Animalutelor % 15:00 18:30 20:00
Cinema 4 % Minionii 2 % 16:00 18:30 20:30
Cinema 1 % Buna dimineata % 09:30
```

- [2,5 p.] Să se memoreze datele din fișier într-o singură structură de date astfel încât să se răspundă cât mai eficient la cerințele de la punctele următoare.
- [1 p.] Scrieți o funcție *sterge\_ore* care are următorii parametri (în această ordine):
  - structura în care s-au memorat datele la cerința a)
  - un șir de caractere *cinema* reprezentând numele unui cinematograf
  - un șir de caractere *film* reprezentând numele unui film
  - mulțime *ore* având ca elemente șiruri de caractere de forma *hh:mm*

Funcția va șterge din programul cinematografului *cinema* programările filmului *film* de la orele din mulțimea *ore* și va returna o listă cu filmele programate la cinematograful *cinema* după această actualizare. Se citește de la tastatură un nume de film *f*, un nume de cinematograf *c* și un șir de caractere *o* de forma *hh:mm* reprezentând o oră. Să se apeleze funcția *sterge\_ore* pentru a șterge programarea filmului *f* la cinematograful *c* la ora *o* și să se afișeze lista returnată; după apelul funcției să se afișeze și structura în care s-au memorat datele.

- c) [1,5 p.] Scrieți o funcție *cinema\_film* care primește următorii parametri: structura în care s-au memorat datele la cerința a), un număr variabil de șiruri de caractere reprezentând nume de cinematografe și doi parametri *ora\_minima* și *ora\_maxima* șiruri de caractere de forma "hh:mm" reprezentând ore. Funcția returnează o listă de tupluri cu elementele de tip (*nume\_film*, *nume\_cinema*, *lista\_de\_ore*) cu filmele care rulează (încep) la cel puțin unul dintre cinematografele primite ca parametru între orele *ora\_minima* și *ora\_maxima*, unde:

- *nume\_film* este numele unui astfel de film
- *nume\_cinema* este un nume de cinema dintre cele primite ca parametru la care rulează filmul *nume\_film*
- *lista\_de\_ore* este lista orelor la care este programat filmul *nume\_film* la cinematograful *nume\_cinema* între orele *ora\_minima* și *ora\_maxima*, ordonată crescător

Lista returnată va fi ordonată crescător după numele filmului, apoi, în caz de egalitate, descrescător după numărul de elemente din *lista\_de\_ore*. Să se apeleze funcția pentru cinematografele 'Cinema 1' și 'Cinema 2', *ora\_minima* "14:00" și *ora\_maxima* "22:00" și să se afișeze lista returnată. *Explicații:* pentru datele din fișier lista returnată va fi [(*Gasca Animalutelor*, 'Cinema 2', ['15:00', '18:30', '20:00']), (*Minionii 2*, 'Cinema 2', ['15:00', '18:30', '20:30']), (*Minionii 2*, 'Cinema 1', ['18:30'])]; filmul 'Elfii cofetari' nu apare în listă deoarece este programat mai devreme de ora "14:00".

### Module (dacă este timp)

Creați modulul *algoritmi\_elementari* care să conțină câte o funcție pentru fiecare dintre următoarele cerințe:

- a) calculează numărul de cifre ale unui număr (recursiv);
- b) returnează inversul unui număr (recursiv)
- c) calculează cmmdc a două numere (recursiv)

Importați modulul *algoritmi\_elementari* și folosiți apeluri ale funcțiilor definite.

Implementați o funcție care primește un număr variabil de numere naturale și un număr *c* și returnează câte dintre numerele întregi primite ca parametru au numărul de cifre cel puțin *c* (folosind funcția de la subpunctul a) din modulul *algoritmi\_elementari*). Apelați funcția pentru numerele 15, 19, 178 și 2856 și cu valoarea parametrului *c* egală cu 2 (parametrul actual corespunzător lui *c* la apel este 2).