

1) a) Scrieți o funcție **min_max** care primește ca parametru o listă nevidă de numere întregi și returnează două valori: valoarea minimă și valoarea maximă din lista primită ca parametru. De exemplu, pentru lista [3, -3, 1, 7, 3, 2] funcția trebuie să returneze valorile -3 și 7. (1 p.)

b) Considerăm un fișier text care conține pe fiecare linie câte un șir format din numere întregi. Scrieți o funcție **incarca_fisier** care să încarce conținutul unui astfel de fișier text într-o listă, astfel: fiecare element al listei va fi o listă conținând toate numerele de pe o linie a fișierului text. Funcția va primi ca parametru numele fișierului text și va returna lista creată. De exemplu, dacă fișierul text este

1 210 7

2 2 2

-3 1 -2 1

9

210 -30 210 10

atunci funcția trebuie să furnizeze lista [[1, 210, 7], [2, 2, 2], [-3, 1, -2, 1], [9], [210, -30, 210, 10]]. Fiecare linie conține cel puțin un număr întreg, iar numerele de pe o linie sunt despărțite între ele prin unul sau mai multe spații

c) Folosind apeluri utile ale funcțiilor definite anterior, scrieți un program care să citească de la tastatură numele unui fișier text de tipul descris mai sus și apoi să rezolve următoarele două cerințe:

- Să scrie în fișierul text *egale.txt* numerele de ordine ale liniilor din fișierul respectiv care conțin doar valori egale între ele. În cazul în care nu există nicio linie cu proprietatea cerută, se va scrie în fișierul *egale.txt* mesajul "Nu există!". Liniile fișierului text de intrare se consideră numerotate de la 0, de sus în jos. De exemplu, pentru fișierul text de mai sus, în fișierul rezultat.txt trebuie scrise, pe două linii, numerele 1 și 3.

- Să afișeze pe ecran două numere întregi a și b din fișierul text dat cu proprietatea că toate numerele din fișierul text aparțin intervalului $[a, b]$. De exemplu, pentru fișierul text de mai sus, trebuie afișate numerele $a = -30$ și $b = 210$.

2) a) Scrieți o funcție **deviruseaza** care primește ca parametru o propoziție modificată de un virus și returnează propoziția corectată: virusul a acționat asupra propoziției inițiale (formată din cuvinte separate prin câte un spațiu) astfel: la fiecare cuvânt a interschimbat prima literă cu ultima, apoi inversat ordinea în care cuvintele apar în propoziție (primul a devenit ultimul, a doilea penultimul etc). De exemplu, pentru propoziția "aorectc aropozitip este aceasta" primită ca parametru funcția va returna "aceasta este propozitia corecta"

b) Scrieți o funcție **prime** care primește ca parametru un număr natural n și un parametru **numar_maxim** cu valoarea implicită 0 care returnează o listă cu numere prime mai mici decât n . Dacă se specifică o valoare nenulă pentru parametrul **numar_maxim**, atunci funcția va returna doar primele **numar_maxim** numere prime mai mici decât n .

c) Se dă fișierul *intrare.in* în care fiecare linie reprezintă o propoziție în care cuvintele sunt separate prin câte un spațiu. Propozițiile din *a* (cest fișier au fost modificate de un virus care acționează așa cum a fost descris la punctul a), dar care a acționat doar asupra liniilor de indice prim din fișier (numerotarea liniilor începând de la 1). Folosind apeluri ale funcțiilor de la a) și b) să se creeze fișierul *intrare_devirusata.out* cu propozițiile din fișierul inițial, dar cu cele de pe poziții prime corectate.

intrare.in	<i>intrare_devirusata.out</i>
prima eropozitip aoud a eropozitip areit a a patra propozitie aincec a a sasea aaptes a is	prima a doua propozitie a treia propozitie a patra propozitie a cincea a sasea si a saptea

3 a) Scrieți o funcție **cifra_control** care primește ca parametru un număr natural n și returnează cifra de control a numărului n . Cifra de control a unui număr natural se obține efectuând suma cifrelor sale, apoi suma cifrelor acestei sume, până când suma obținută este un număr format dintr-o singură cifră.

b) Scrieți o funcție **insereaza_cifra_control** care primește ca parametru o listă de numere naturale și o modifică inserând după fiecare element din listă cifra sa de control (!funcția modifică lista primită ca parametru, nu returnează lista modificată). Cifra de control a unui număr se va calcula folosind funcția de la a)

c) Scrieți o funcție **egale** care primește ca parametri un număr variabil de liste cu elemente numere naturale și întoarce True dacă listele primite ca parametru sunt egale și False altfel.

d) Se consideră fișierul *numere.in* care conține numere naturale separate prin spații. Folosind funcția de la b) afișați pe ecran fiecare număr din fișier urmat de cifra sa de control. După fiecare cifră de control afișată se va trece la linie nouă

numere.in	Iesire pe ecran
10 78 8051 91	10 1 78 6 8051 5 91 1

e) Se dau două fișiere *numere.in* și *numere2.in* conținând numere naturale separate prin spații. Asociem fiecărui fișier un șir de cifre de control astfel: considerăm numerele distincte din fișier în ordine crescătoare și formăm un șir cu cifrele lor de control (în această ordine). Spunem că cele două fișiere sunt similare dacă șirurile asociate celor două fișiere sunt egale. Folosind funcțiile de la punctele b și c să se determine dacă cele două fișiere sunt similare și să se afișeze pe ecran un mesaj corespunzător da/nu

numere.in	numere2.in
100 10 78 8051 10 110	51 100 200 788 100 19 51

Numerele distincte din primul fișier în ordine crescătoare sunt: 10, 78, 100, 110, 8051
Șirul cifrelor de control asociat primului fișier va fi atunci: 1, 6, 1, 2, 5

Numerele distincte din al doilea fișier în ordine crescătoare sunt: 19, 51, 100, 200, 788
Șirul cifrelor de control asociat primului fișier va fi atunci: 1, 6, 1, 2, 5
Cele două șiruri sunt egale, deci se va afișa da.

4) Spiridușii lui Moș Crăciun au fiecare asociat câte un cod format din litere și cifre. Până în luna noiembrie spiridușii au avut la dispoziție un fișier în care să completeze ce jucării pot face până la Crăciun și câte bucăți. O linie din acest fișier conține codul spiridușului, numărul de bucăți (număr natural) și numele jucăriei (numele este format din cuvinte separate prin câte un spațiu). Un spiriduș poate adăuga de mai multe ori o linie în fișier, chiar și cu aceeași jucărie, dacă se hotărăște că poate face mai multe.

Un exemplu de fișier este:

```
S1 1 papusa  
S2 1 papusa  
S3 1 masinuta  
S1 10 trenulet  
S2 1 papusa  
S2 2 masinuta  
S1 10 ponei  
S3 15 ponei
```

a) Memorați datele din fișier astfel încât Moș Crăciun să poată afla cât mai repede informațiile cerute la punctele următoare.

b) Dat codul unui spiriduș, care sunt jucăriile pe care le poate face și ce cantitate din fiecare? Pentru aceasta scrieți o funcție **despre_spiridus** cu 2 parametri: în primul parametru se transmite structura în care s-au memorat datele la punctul a) iar al doilea este codul unui spiriduș. Funcția returnează o listă cu elementele tupluri de 2 elemente – primul fiind numele jucăriei, iar al doilea cantitatea – ordonată descrescător după cantitate și, în caz de egalitate, crescător după nume. Apelați funcția pentru codul S1 și afișați lista returnată de funcție. Pentru datele din fișierul exemplu se va afișa [('ponei', 10), ('trenulet', 10), ('papusa', 1)]

c) Care este mulțimea jucăriilor pe care spiridușii le pot produce? Pentru aceasta scrieți o funcție **jucarii** care primește ca parametru structura în care s-au memorat datele la punctul a) și returnează o mulțime cu numele jucăriilor care pot fi produse de spiriduși. Apelați funcția și afișați pe ecran elementele mulțimii returnate (pe o linie, separate prin virgula). Pentru datele din fișierul exemplu o posibilă ieșire este (jucăriile se pot afișa în orice ordine): **ponei, trenulet, masinuta, papusa**

d) Care este lista spiridușilor harnici: ordonați descrescător după numărul de jucării diferite pe care le pot face și, în caz de egalitate, descrescător după cantitatea de jucării pe care o vor produce și, în caz de egalitate, crescător după cod? Pentru aceasta scrieți o funcție **spiridusi** care primește ca parametru structura în care s-au memorat datele și returnează o listă cu elementele tupluri de 3 elemente – primul fiind codul spiridușului, al doilea numărul de jucării diferite pe care le poate produce, iar al treilea numărul total de bucăți de jucării care le poate produce spiridușul – ordonată după criteriile cerute de Moș Crăciun (precizate anterior). Apelați funcția și afișați pe ecran elementele listei obținute, fiecare tuplu din listă fiind afișat pe o linie separată. Pentru datele din fișierul exemplu se va afișa

('S1', 3, 21)

('S3', 2, 16)

('S2', 2, 4)

e) În caz că un spiriduș nu mai poate produce un tip de jucărie, să se actualizeze informațiile asociate spiridușului. Pentru aceasta scrieți o funcție **actualizare** care primește 3 parametri (în aceasta ordine): structura în care s-au memorat datele, codul spiridușului și numele jucăriei. Dacă spiridușul dat ca parametru produce cel puțin două tipuri de jucării, funcția va șterge din structura de date informațiile despre jucăria cu numele dat ca parametru atașate spiridușului cu codul dat ca parametru și va returna **True**. Altfel funcția va returna **False**. Să se apeleze

funcția pentru spiridușul cu cod S1 și jucăria trenulet, apoi să se afișeze informațiile rămase despre S1 folosind funcția **despre_spiridus** de la punctul b. Pentru datele din fișierul exemplu se va afișa [('ponei', 10), ('papusa', 1)]