

0. Problemele de la seminar

1. Scrieți un program care să determine grupurile de cuvinte dintr-un fișier text care p-rimează între ele = au aceleași ultime p-litere (p citit de la tastatura). Numele fișierului de intrare se va citi de la tastatură, iar grupurile se vor scrie în fișierul text "rime.txt", câte un grup pe o linie, *în ordine descrescătoare în raport cu numărul de elemente din grup*. Cuvintele din fiecare grup vor fi *sortate lexicografic descrescător*.

De exemplu, pentru $p=2$ și fișierul:

```
ana dana  
mere pere prune  
bune  
banana si gutui amare are
```

rime.txt va fi:

```
pere mere are amare  
dana banana ana  
prune bune  
si  
gutui
```

2. Din fișierul **studenti.in** se citesc, de pe câte o linie, date despre câte un student, astfel:
Nume, Prenume, grupa, nota1, nota2, ... (numărul de note poate varia între 0 și 5, deoarece în sesiune sunt 5 examene și la unele examene studentul poate fi absent).
 - a) Să se afișeze pentru fiecare grupă în parte studenții;
 - b) Să se afișeze pentru o grupă citită de la tastatură numele și media fiecărui student dacă a fost prezent și a promovat toate cele 5 examene din sesiune sau mesajul "restantier"
 - c) Pentru o eventuală repartizare a locurilor de cazare, să se afișeze toți studenții ordonați astfel:
 - mai întâi integraliștii ordonați după medie descrescător;
 - restanțierii ordonați după numărul de restanțe crescător.

Exemplu de fișier de intrare:

```
Petre, Ana, 191, 9, 10, 8, 8, 5  
Marin, Ioana, 191, 5, 10, 10, 7, 8  
Popescu, Maria, 191, 10, 10, 10  
Marin, Ioana, 191, 5, 4, 5, 8, 8  
Zaru, Anton, 191, 10, 10, 8, 8, 9  
Petrecu, Mircea, 192, 10, 10, 10, 10, 10
```

La punctul b pentru grupa 191 se va afișa

```
Petre Ana 8.0  
Marin Ioana restantier  
Popescu Maria restantier  
Zaru Anton 9.0
```

La punctul c se va afișa

```
Petrecu Mircea media 10.0  
Zaru Anton media 9.0  
Petre Ana media 8.0  
Marin Ioana restante 1  
Popescu Maria restante 2
```

3. a) Scrieți o funcție care primește ca parametru două nume de fișiere (*variantă: un număr variabil de nume de fișiere*) și returnează un dicționar cu cuvintele care apar în cel puțin unul dintre fișiere și frecvența totală cu care apare fiecare cuvânt (suma frecvențelor cu care apar în fișiere). Cuvintele pot fi pe mai multe linii și pe o linie sunt separate prin spații.
- b) Se consideră fișierele cuvinte1.in si cuvinte2.in. Să se afișeze cuvintele care apar în cel puțin unul dintre fișiere ordonate crescător lexicografic
- c) Se consideră fișierul cuvinte1.in. Să se creeze o listă de perechi (cuvinte, frecvențe) cu cuvintele care apar în fișier și frecvența cu care apar, *ordonată descrescător după frecvență* (folosind funcția de la a)).
- d) Să se determine un cuvânt care apare cel mai des în cuvinte2.in, folosind funcția de la a) și funcția max. Dacă sunt mai multe posibilități, se va afișa cuvântul cel mai mic din punct de vedere lexicografic
- e) Pentru două documente text, F_1 și F_2 , și $\{c_1, c_2, \dots, c_n\}$ mulțimea cuvintelor care apar în cel puțin unul din cele două documente. Pentru $1 \leq i \leq n$, fie v_{i1}, v_{i2} numărul de apariții al cuvântului i în primul, respectiv în al doilea document. Distanța cosinus dintre cele două documente, notată $dcos(F_1, F_2)$, dintre F_1 și F_2 se calculează după formula:

$$dcos(F_1, F_2) = \frac{\sum_{i=1}^n v_{i1} v_{i2}}{\sqrt{\sum_{i=1}^n v_{i1}^2} \sqrt{\sum_{i=1}^n v_{i2}^2}}$$

Să se calculeze distanța cosinus dintre fișierele caractere1.in si caractere2.in (folosind funcția de la a) apelată separat pentru fiecare fișier) cu două zecimale.

Exemplu Pentru următoarele două fișiere

cuvinte1.in	cuvinte2.in
a fost	a fost
a fost	a fost
odata ca in povesti	altadata ca in basme
o data ca in povesti	alta data ca in basme
in povesti	nu in povesti

se va afișa:

- a) dicționarul va fi {'a': 4, 'fost': 4, 'odata': 1, 'ca': 4, 'in': 6, 'povesti': 4, 'o': 1, 'data': 2, 'altadata': 1, 'basmе': 2, 'alta': 1, 'nu': 1}
- b) a alta altadata basme ca data fost in nu o odata povesti
- c) [('in', 3), ('povesti', 3), ('a', 2), ('fost', 2), ('ca', 2), ('odata', 1), ('o', 1), ('data', 1)]
- d) in
- e) 0.79

4. În fișierul text “test.in” se află testul unui elev de clasa a II-a la matematică, conținând 9 înmulțiri scrise pe rânduri distincte. Un calcul corect este notat cu un punct, iar unul incorect cu 0 puncte. Să se realizeze un program care să evalueze testul dat, astfel: în dreptul fiecărui calcul corect se va scrie mesajul ‘Corect’, iar în dreptul fiecărui calcul greșit se va scrie mesajul ‘Gresit’ și rezultatul corect, iar la final se va scrie nota (un punct se acordă din oficiu!). Rezultatul evaluării testului se va afișa în fișierul test.out

test.in	Test.out
3*4=11	3*4=11 Gresit 12
2*10=20	2*10=20 Corect
5*5=24	5*5=24 Gresit 25
7*4=28	7*4=28 Corect
2*6=12	2*6=12 Corect
10*10=100	10*10=100 Corect
3*9=27	3*9=27 Corect
6*7=33	6*7=33 Gresit 42
0*9=1	0*9=1 Gresit 0
	Nota 6

5. Se citesc de la tastatură două numere naturale n și m .
- a) Să se scrie în fișierul matrice.in o matrice cu n linii și m coloane cu elemente mai mici decât 100 generate aleator (fără a o memora). Exemplu de generare a unui număr mai mic decât 100:

random.randrange(1,100)

- b) Se citește din fișierul matrice.in matricea generata la a). Să se genereze în memorie și să se scrie în fișierul matrice.out transpusa matricei și matricea obținută din matricea inițială ordonând crescător elementele de pe ultima coloana prin interschimbări de linii.
6. În fișierul text “date.in” sunt memorate, pe linii, numele și prenumele studenților dintr-o grupă. Să se scrie un program care să genereze conturile de email ale studenților și parolele temporare, după care să le salveze în fișierul text “date.out”. Contul de email al unui student va fi de forma prenume.numa@s.unibuc.ro, iar parola temporară va fi de forma o literă mare, 3 litere mici și 4 cifre. Se va scrie o funcție care generează parola folosind funcții din modulul random <https://docs.python.org/3/library/random.html> (randint, choice, choices pentru constantele string.ascii_uppercase, string.digits etc din modulul string <https://docs.python.org/3/library/string.html>). Exemplu de generare de 3 litere mici:

random.choices(string.ascii_letters , k=3)

date.in

Boboceca Andrei

Marinescu Ciprian

Vasile Dragos

date.out (exemplu, parolele sunt generate aleator)

andrei.bobocea@s.unibuc.ro, Wadf2133

ciprian.marinescu@s.unibuc.ro, Qsdd2111

dragos.vasile@s.unibuc.ro, Bbyt7690

7. În fișierul "elevi.in" sunt memorate informații despre elevii unei clase; astfel, pe o linie a fișierului se dau următoarele informații despre un elev: cnp, nume (fără spații), prenume (fără spații), lista de note, de exemplu:

2501910000034 Ionescu Ion 10 8 7 8

2402900000041 Marinica Maria 9 10 8 8 8

1412900000041 Petrescu Petrica 8 10 4 7

a) Memorați lista de elevi din fișier astfel încât să se poată răspundă cât mai eficient la întrebări de tipul celor de la subpunctele următoare (dat cnp elev, care sunt numele, notele, să se lista de note a elevului).

b) Scrieți o funcție care primește ca parametri un cnp și structura de date în care s-au memorat datele despre elevi la punctul a) și crește cu 1 prima notă a elevului cu cnp-ul primit ca parametru. Funcția returnează nota după modificare sau None dacă cnp-ul nu există. Apelați funcția pentru un cnp citit de la tastatură.

c) Scrieți o funcție care primește ca parametri un cnp, o listă de note și structura de date în care s-au memorat datele despre elevi la punctul a) și adaugă lista de note la notele elevului cu cnp-ul primit ca parametru. Funcția returnează lista de note după modificare sau None dacă cnp-ul nu există. Apelați funcția pentru un cnp citit de la tastatură și lista l_note=[10,8].

d) Scrieți o funcție care primește ca parametri un cnp și structura de date în care s-au memorat datele despre elevi la punctul a) și șterge informațiile despre elevul cu acest cnp. Apelați funcția pentru un cnp citit de la tastatură (dacă cnp-ul nu este în listă funcția nu va modifica nimic și nu va da eroare)

e) Folosind structura de date în care s-au memorat datele despre elevi la punctul a) (nu citind din nou datele) construiți în memorie o lista de liste cu elevii din fișier, un element din lista fiind de forma [nume, prenume, lista de note], ordonată **descrescător** după medie și, în caz de egalitate, după nume și afișați elementele listei în fișierul „elevi.out”.

f) Scrieți o funcție care primește ca parametru structura de date în care s-au memorat datele despre elevi la punctul a) și adaugă la informațiile asociate unui student un cod de lungime 6 generat aleator care conține 3 litere urmate de 3 cifre. Exemplu de apel:

```
genereaza_coduri(d)
```

```
print(d)
```