

0. Probleme rămase din seminar 4

DICTIONARE

1. a) Scrieți o funcție care primește ca parametru 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 și 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 și 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

2. 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

FISIERE

3. 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 scrie în fișierul text “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

4. 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>)

date.in

Bobocia Andrei

Marinescu Ciprian

Vasile Dragos

date.out (exemplu, parolele sunt generate aleator)

andrei.bobocia@s.unibuc.ro,Wadf2133

ciprian.marinescu@s.unibuc.ro,Qsdd2111

dragos.vasile@s.unibuc.ro,Bbyt7690

FUNCTII

5. Scrieți o funcție `negative_pozitive` care primește ca parametru o lista de numere întregi și returnează două liste, prima formată din numerele strict negative din lista primită ca parametru și a doua formată din cele strict pozitive. Se dă un fișier `numere.in` care conține numere întregi pe mai multe linii, separate prin spații. Să se creeze în memorie o listă `ls_numere` cu numerele din fișier. Folosind apeluri utile ale funcției `negative_pozitive` să se scrie la sfârșitul fișierului text al cărui nume va fi citit de la tastatură(deschis pentru adăugare cu modul “a”), pe două linii separate, numerele strict negative, respectiv cele strict pozitive, ordonate crescător (folosind `write` sau `writelines`) din `ls_numere`.
6. 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 listă 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)
```

Parametri default

7. 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.

Număr variabil de parametri

8. Scrieți o funcție care primește un număr întreg x și un număr variabil de liste nevide de numere întregi și returnează numărul de liste primite ca parametru care conțin x.

Exemplu apel:

```
nr = liste_x(3, [1, 5, 7], [3], [1, 8, 3], [])
print(nr)
```

b) Modificați funcția de la a astfel încât rezultatul să nu fie returnat, ci să se salveze în variabila globală **rez**. Exemplu apel:

```
rez = None
liste_x(3, [1, 5, 7], [3], [1, 8, 3], [4,3])
```

`print(rez)`

9. 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.
10. 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!... Totus este trist in lume!	Aceesi 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