Laborator 6 PA - DICȚIONARE

https://docs.python.org/3/library/stdtypes.html#dict

https://docs.python.org/3/library/random.html

https://docs.python.org/3/library/string.html

1. Sortarea unui vector după un criteriu - parametrul key

SORTARI LISTE

a) Se consideră o listă de numere. Să se sorteze folosind o funcție lambda așa cum s-ar fi sortat dacă numerele erau șiruri.

b) Se consideră o listă de numere. Să se sorteze folosind o funcție lambda comparând întâi ultima cifră, apoi penultima, etc. (așa cum am fi sortat dacă erau șiruri cu literele în ordine inversă).

c) Se consideră o listă de numere. Să se sorteze descrescător după lungimea numărului.

d) Se consideră o listă de numere. Să se sorteze după numărul de cifre distincte.

e) Sortare numere întâi după lungime și apoi descrescător după valoare.

TEMA:

f) Să se sorteze o listă de numere naturale astfel încât numerele pare sortate descrescător să fie poziționate după cele impare sortate crescător.
g) Se citește o propoziție cu cuvinte separate prin spațiu. Să se formeze o nouă propoziție cu cuvintele din prima propoziție care au lungime cel puțin 2 ordonate descrescător după lungime.
h) Se citește un vector de numere naturale (cu elementele date pe o linie, separate prin spațiu). Să se ordoneze elementele din vector crescător după suma cifrelor, iar în caz de egalitate, descrescător după valorile lor
v = [11, 45, 20, 810, 179, 81, 1000] => v = [1000, 20, 11, 810, 81, 45, 179]

2. 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 si 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)
```

3. 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
```

4. Se dă un fișier cu cuvinte pe mai multe linii separate prin spații. Scrieți un program care să determine grupurile de cuvinte din fișier care au aceleași litere (nu neapărat cu aceeași frecvență). *Numele fișierului de intrare se va citi de la tastatură*, iar grupurile formate din cel puțin două cuvinte se vor scrie în fișierul text "litere.txt", câte un grup pe o linie. Cuvintele din fiecare grup vor fi sortate după lungime, **iar în caz de lungimi egale, lexicografic**, iar *grupurile se vor scrie în fișier în ordinea descrescătoare a numărului de elemente din mulțimile literelor*.

Pentru fișierul de intrare:

```
apar mare
si amara  rapa para
par isi rama
```

fișierul de ieșire va fi:

```
par apar para rapa
rama amara
si isi
```

5. 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 $\{c_1, c_2, ..., c_n\}$ 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 odata ca in povesti o data ca in povesti in povesti	a fost a fost altadata ca in basme alta data ca in basme 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, 'basme': 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

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.nume@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

Bobocea 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. Problemele rămase de la seminar