**PROBLEME**

✓ 1. Se citește din fișierul "numere.in" un șir de cuvinte separate prin spațiu. Unele dintre cuvinte sunt numere naturale. Să se afișeze în fișierul “numere.out” media aritmetică a cuvintelor din șir care sunt numere naturale. Mai exact, rezultatul va fi un șir de forma: “(nr1+...nr\_k)/n=media”, unde nr1,...,nrk sunt numerele care apar în șir și media este afișată cu 2 zecimale

f = open("numere.in")  
m = 0  
nr = 0  
l = []  
ls = [x for x in f.readline().split()]  
for i in ls:  
 if i.isnumeric() == True:  
 m = m + int(i)  
 nr = nr + 1  
 l.append(i)  
for i in range(len(l)):  
 print (l[i], end=" ")  
 if i != len(l)-1:  
 print ("+", end=" ")  
print (f"= {m/nr:.2f}") # impartire exacta

✓ 2. Fișierul text “numere\_comune.in” conține numere naturale despărțite prin spații și scrise pe mai multe linii.

Să se afișeze în fișierul “comune.out” numerele care apar pe toate liniile din fișier ordonate crescator (folosind set + intersectie).

**Exemplu:** dacă fișierul “numere\_comune.in” conține

2 1 5 1 3

1 4 2 2

2 1 1 6 8

atunci fișierul “comune.out” va conține (numerele nu neapărat în această ordine):

1 2

f = open ("numere\_comune.in")  
g = open ("numere\_comune.out","w")  
ls = [set(linie) for linie in f]  
m = ls[0] # prima multime  
for i in ls:  
 m = m.intersection(i) # intersectia lui m cu i  
m.remove(" ")  
m = list(m)  
m.sort()  
g.write (str(m))

✓ 3. Se citesc din fișierul “puncte.in” informații despre puncte în plan: cele două coordonate (numere întregi) și eticheta (care poate fi un șir ce conține și spatii). Mai exact, structura fișierului este următoarea: pe fiecare linie a fișierului sunt date informațiile despre un punct: doua coordonate întregi si eticheta punctului.

1 2 punctul 1  
1 3 punctul 2  
2 5 punctul 3  
1 2 punctul 1 nou  
4 1 punctul 4

Asupra punctelor din fișier se fac operațiile codificate in fișierul “interogari.in” astfel: pe fiecare linie se dau doua numere reprezentând coordonatele unui punct si o valoare din mulțimea {1,0} cu semnificația interogare (1)/ștergere(0).

1 2 1  
2 4 1  
1 2 0  
1 3 0  
4 1 1  
10 20 1  
20 40 0

Ca rezultat al operațiilor se va crea fișierul interogari.out în care în dreptul punctelor cu 1 se va scrie eticheta (sau mesajul *nu exista*), iar la final se va afișa mesajul: “punctele ramase” și apoi vor afișa punctele rămase după ștergerea punctelor cu 0 (sub forma (x,y): eticheta). Dacă un punct apare de mai multe ori în puncte.in se păstrează ultima etichetă asociată lui.

(1,2) punctul 1 nou  
(2,4) nu exista  
(4,1) punctul 4  
(10,20) nu exista  
punctele ramase  
(2, 5): punctul 3  
(4, 1): punctul 4

f = open ("puncte.in")  
d = {}  
for linie in f:  
 ls = linie.split(maxsplit=2)  
 x = int(ls[0])  
 y = int(ls[1])  
 et = ls[2].rstrip("\n")  
 t = (x,y)  
 d[t] = et  
f.close()  
print(d)  
f = open ("interogari.in")  
g = open ("interogari.out","w")  
for linie in f:  
 ls = linie.split()  
 x = int(ls[0])  
 y = int(ls[1])  
 t = (x,y)  
 if int(ls[2]) == 1:  
 if t in d:  
 g.write(f"{t} {d[t]} \n")  
 else:  
 g.write(f"{t} nu exista \n")  
 else:  
 if t in d:  
 del d[t]  
g.write("puncte ramase:\n")  
for t in d:  
 g.write(f"{t} {d[t]} \n")  
f.close()  
g.close()

✓ 4. a) Scrieți o funcție care, dat numele unui fișier, determină și returnează frecvența caracterelor din fișier

b) Se consideră fișierele caractere1.in si caractere2.in. Să se afișeze pentru fiecare fișier frecvența caracterelor.

c) Să se afișeze caracterele comune celor două fișiere și frecvența cu care se repetă în ambele fișiere (se vor afișa întâi caracterele alfanumerice ordonate crescător, apoi celelalte).

def frecventa (nume\_fisier):  
 frecv = {} # dictionar  
 f = open (nume\_fisier)  
 for line in f:  
 for i in line:  
 if i in frecv:  
 frecv[i] = frecv[i] + 1  
 else:  
 frecv [i] = 1  
 f.close()  
 return frecv  
print(frecventa ("caractere1.in"))  
print(frecventa ("caractere2.in"))  
f1 = frecventa ("caractere1.in")  
f2 = frecventa ("caractere2.in")  
d = {}  
for i in f1:  
 if i in f2:  
 d[i] = min(f1[i],f2[i])  
print (d)

✓ 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).

b) Se citește din fișierul matrice.in matricea generata la a). Să se genereze în memorie li 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.

import random  
f = open ("matrice.in","w")  
n = int(input()) #linii  
m = int(input()) #coloane  
for i in range (n):  
 for j in range (m):  
 f.write (str(random.randint(0,100))) # numere random intre 0 si 99  
 f.write (" ")  
 f.write ("\n")  
f.close()  
f = open ("matrice.in")  
g = open ("matrice.out", "w")  
l = []  
l = [ line.split() for line in f] # citire matrice din fisier  
for j in range (m): # transpusa  
 for i in range (n):  
 g.write (l[i][j])  
 g.write (" ")  
 g.write ("\n")  
g.write("\n")  
for i in range (n):  
 for j in range (i+1,n):  
 if l[i][m-1]>l[j][m-1]:  
 aux = l[i]  
 l[i] = l[j]  
 l[j] = aux  
for i in range (n):  
 for j in range (m):  
 g.write (l[i][j])  
 g.write (" ")  
 g.write ("\n")

6. Se dă un fișier cu cuvinte separate prin spatii. 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**.

Pentru fișierul de intrare:

apar mare  
si amara rapa para  
par isi rama

fișierul de ieșire va fi

apar mare  
si amara rapa para  
par isi rama

**nume\_fisier=input()**   
**d={}**   
**with open(nume\_fisier) as f:**   
**for linie in f:**   
**ls=linie.split()**   
**for c in ls:**   
**fs=frozenset(set(c))**   
**d.setdefault(fs,[]).append(c)**   
   
**g=open("litere\_egale.txt","w")**   
**#g.write(str(d))**   
**for lista in d.values():**   
**if len(lista)>=2:**   
**g.write(" ".join(sorted(lista,key=lambda x: (len(x),x))))**   
**g.write("\n")**   
**g.close()**

7. Se consideră un fișier de intrare produse.in cu informații despre magazine, produse și cantitatea de produse din fiecare magazin, sun forma:

**Magazin** cod\_magazin nume\_magazin

cantitate nume\_produs

...

cantitate nume\_produs

**Magazin** cod\_magazin nume\_magazin

cantitate nume\_produs

...

cantitate nume\_produs

Exemplu:

Magazin 123 magazin1  
5 mere  
7 pere  
2 prune  
Magazin 221 magazin 2  
3.5 pere  
10 banane

a) Să se memoreze datele astfel încât să răspundă la interogări de tipul: dat codul magazinului și numele unui produs, să se afișeze cantitatea de produs din magazin (să se interogheze pentru un cod și un nume date de la tastatură), dat codul magazinului, numele unui produs și o cantitate de produs care se vinde, actualizează stocul de produs din magazin dacă se poate (să se facă o astfel de actualizare pentru cod, nume produs, cantitate date de la tastatură)

b) Să se afișeze o lista de tupluri (magazin, lista de (produse:cantitate)) ordonată după cantitatea totală de marfă din magazin, și, în caz de egalitate, după numele magazinului

c) Să se afișeze o listă a tuturor produselor care se găsesc în magazine (folosind reuniune de mulțimi)

**f=open("magazin.in"); cod=0; d={}**   
**for linie in f:**   
**ls=linie.split()**   
**if ls[0]=="Magazin":**   
**cod=int(ls[1]); nume\_magazin=" ".join(ls[2:])**   
**d[cod]=(nume\_magazin, {})**   
**else:**   
**cantitate=float(ls[0]);    nume\_produs=" ".join(ls[1:])**   
**d[cod][1][nume\_produs]=cantitate**   
**f.close()**   
**print(d)**   
**cod\_magazin=221; nume\_produs="asd"**   
**print(d[cod\_magazin][1].get(nume\_produs,"nu exista produsul"))**   
**cod\_magazin=221; nume\_produs="asd"**   
**cantitate=3.5**   
**if d[cod\_magazin][1].get(nume\_produs,0)>=cantitate:**   
**d[cod\_magazin][1][nume\_produs]=d[cod\_magazin][1][nume\_produs]-cantitate**   
**if d[cod\_magazin][1][nume\_produs]==0:**   
**d[cod\_magazin][1].pop(nume\_produs)**   
**else:**   
**print("stoc indisponibi")**   
**lista\_valori=d.values()**   
**print(sorted(lista\_valori, key=lambda x: (sum(x[1].values()),x[0])))**   
**s=set()**   
**for magazin in d.values():**   
**s1=magazin[1].keys()**   
**s=s|s1**   
**print(s)**