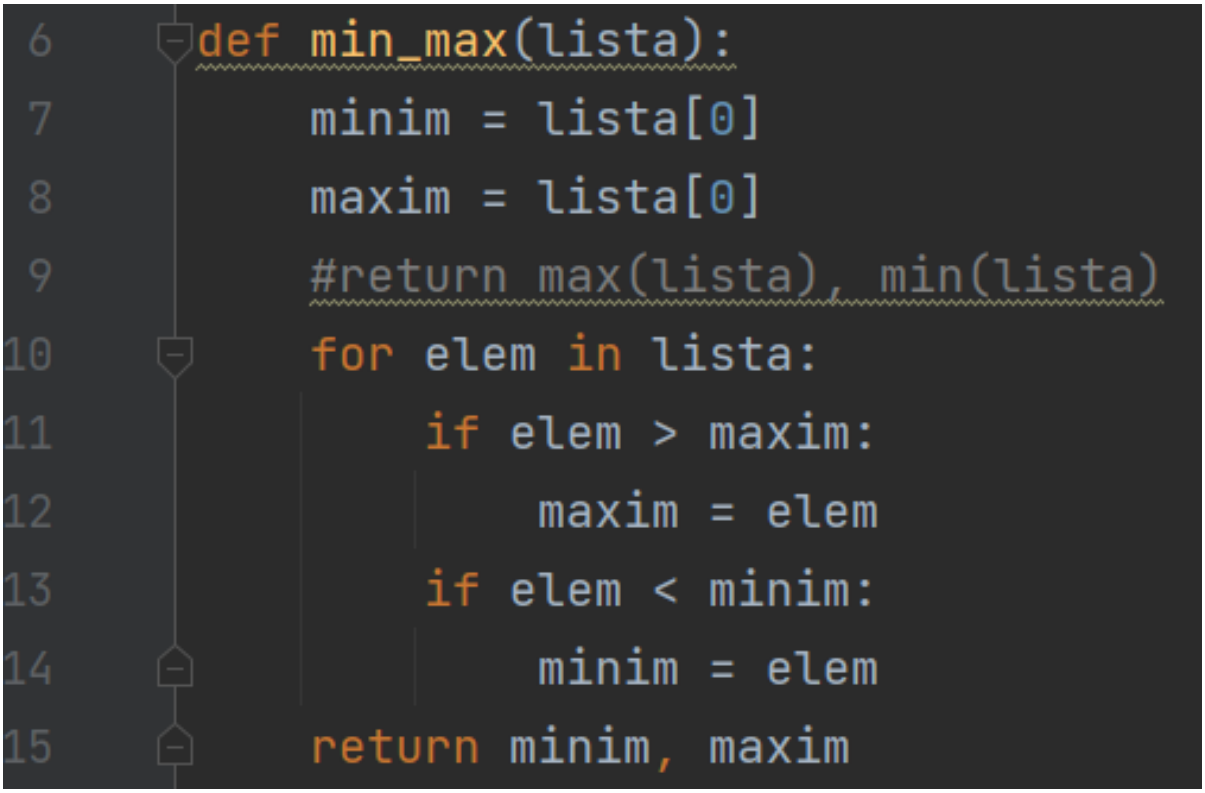
1)

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.)



```
6   def min_max(lista):
7       minim = lista[0]
8       maxim = lista[0]
9       #return max(lista), min(lista)
10      for elem in lista:
11          if elem > maxim:
12              maxim = elem
13          if elem < minim:
14              minim = elem
15      return minim, maxim
```

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, daca fișierul text este 1 210 7 2 2 2 -3 1 -2 1 9 210 -30 210 10 atunci funcția trebuie sa 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

```
35  def incarca_fisier(nume_fisier):
36      f = open(nume_fisier, "r")
37      lista_mare = []
38      linie = f.readline()
39      while linie != '':
40          lista_linie = [int(x) for x in linie.split()] #conversie lista
41          lista_mare.append(lista_linie) #adauga lista in lista mare
42          linie = f.readline()
43      return lista_mare
44      f.close()
```

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.

```
g = open("egale.txt", "w")
nume_fisier_citit = input()
liste_fisier = incarca_fisier(nume_fisier_citit)
afisat = False
for i in range(len(liste_fisier)):
    minimul, maximul = min_max(liste_fisier[i])
    if minimul == maximul:
        g.write(str(i) + "\n")
        afisat = True

if afisat == False:
    g.write("Nu exista!")
```

• 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.

```
nume_fisier_citit = input()
liste_fisier = incarca_fisier(nume_fisier_citit)
afisat = False
minimul_global = liste_fisier[0][0]
maximul_global = liste_fisier[0][0]
for i in range(len(liste_fisier)):
    minimul, maximul = min_max(liste_fisier[i])
    if minimul < minimul_global:
        minimul_global = minimul
    if maximul > maximul_global:
        maximul_global = maximul
print(minimul_global, maximul_global)
```

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” primita ca parametru funcția va returna “aceasta este propozitia corecta”

```
def deviruseaza(prop):

    listaCuv = prop.split()

    listaCuvCorectate = []

    for cuv in listaCuv:
        cuvCorect = cuv[-1] + cuv[1:len(cuv)-1] + cuv[0]

        listaCuvCorectate.append(cuvCorect)

    listaCuvCorectate.reverse()

    return " ".join(listaCuvCorectate)


print(deviruseaza("aorectc aropozitip este aceasta"))
```

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.

```python
def prime(n, numar_maxim = 0):

    primes = []

    for i in range(2,n):

        estePrim = True

        for j in range(2,i):
            if i%j==0:
                estePrim=False
                break

        if estePrim:
            primes.append(i)

    if numar_maxim==0:
        return primes

    return primes[:numar_maxim]

print(prime(20,5))
```

c) Se dă fișierul intrare.in în care fiecare linie reprezintă o propoziție în care cuvintele sunt separate prin câte un spațiu. Propozițiile din acest 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) si b) să se creeze fișierul intrare_devirusata.out cu propozițiile din fișierul inițial, dar cu cele de pe poziții prime corectate

.

```
fin = open("intrare.in","r")
fout = open("intrare_devirusata.out","w")

lista_propozitii = []

for line in fin.readlines():
    lista_propozitii.append(line.strip())

lista_prime = prime(len(lista_propozitii)+1)

for pos in lista_prime:
    # suntem pe o pozitie prima pos
    # noi o sa devirusam lista_propozitii[pos-1], pentru ca lista este indexata de la 0
    lista_propozitii[pos-1] = deviruseaza(lista_propozitii[pos-1])


for prop in lista_propozitii:
    print(prop,file=fout)

fin.close()
fout.close()
```