1 Scrieți o funcție care sortează o lista de numere folosind: mergsort. Subiect eliminatoriu (1p)

2 Specificați si testați următoarea funcție (2p):

```
def f(l):
    if l==None or l==[]: raise ValueError()
    aux = l[0]+1
    for e in l:
        if (e-aux >=0):return False
        aux = e
    return True
```

3 Analizați complexitatea timp si spațiu a următorului algoritm. (2p).

```
def g(n):
   l = []
   for i in range(n*n):
      k = 1
      while k<n:
         l.append(k)
         k=k*2
```

4 Folosind metoda Divide et impera scrieți o funcție pură care calculează numărul de numere negative într-o lista de numere. Datele se împart in 2 părți egale la fiecare pas (2p).

5 Se dă o listă de numere naturale a1,a2,..,an un număr k și un S. Generați toate aranjamentele luate câte k unde suma elementelor este <=S Ex. [15, 8, 12, 3] k=2 și S=23 soluțiile: (15, 8), (8,15) ,(8,12,3),(12,8,3)... Descrieți soluția Backtracking formalizat (candidat, consistent, soluție fără implementare) (2p)

Obs: Subiectele se rezolva pe foaie, scris de mana.

Fiecare pagina, in coltul din dreapta sus, sa conțină: nume prenume, grupa, numărul subiectului, numerotare pagina.

Subiectele se pot rezolva in orice ordine pe foaie.

Nu trebuie sa copiat enunțul problemei (doar sa indicați clar numărul problemei rezolvate)

Daca nu se rezolva subiectul eliminatoriu (problema 1) examenul scris este picat.

Înainte de expirarea timpului trebuie sa trimiteți un singur fișier pdf, care conține poze de pe fiecare pagina de rezolvare. Pozele sa fie cat mai clare, sa aibă orientarea corecta in pdf, o poza per pagina de pdf.

Se corectează doar paginile trimise corect care se pot citi si au fost trimise pana la timpul limita anunțat.