

1 Scrieți o funcție care implementează sortarea prin inserție. Subiect eliminatoriu. (1p)

2 Specificați și testați următoarea funcție (2p):

```
def f2(n):  
    if n<=0: raise ValueError()  
    l = [x for x in range(n-1,-1,-1)]  
    for i in range(n-1):  
        l[i+1] += l[i]  
    return l[-1]
```

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

```
def f(l):  
    if len(l) == 1:  
        return l[0]  
    if l[0]==0:  
        return 0  
    return l[0] * f(l[1:])
```

4 Folosind metoda Divide et impera scrieți o funcție pură care calculează numărul de numere pozitive într-o lista de numere. Datele trebuie împărțite în 2 părți egale la fiecare pas. (2p).

5 Generați toate sublistele unei liste date, cu proprietatea ca sublistele conțin ori doar numere negative ori doar numere pozitive. Descrieți schematic soluția (candidat, consistent, soluție) bazată pe metoda Backtracking (fără implementare) (2p)

Obs: Subiectele se rezolvă pe foaie, scris de mână.

Fiecare pagină, în colțul din dreapta sus, să conțină: nume prenume, grupa, numărul subiectului, numerotare pagină.

Subiectele se pot rezolva în orice ordine pe foaie.

Nu trebuie să copiați enunțul problemei (doar să indicați clar numărul problemei rezolvate)

Dacă nu se rezolvă subiectul eliminatoriu (problema 1) examenul scris este picat.

Înainte de expirarea timpului trebuie să trimiteți un singur fișier pdf, care conține poze de pe fiecare pagină de rezolvare. Pozele să fie cât mai clare, să aibă orientarea corectă în pdf, o poză pe pagină de pdf.

Se corectează doar paginile trimise corect care se pot citi și au fost trimise până la timpul limitat anunțat.