

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

2 Specificați și testați următoarea funcție (2p):

```
def f(l):  
    if l==None or l==[]: raise ValueError()  
    aux = {0:0,1:0}  
    for e in l:  
        aux[e % 2] += 1  
    return aux[0]
```

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

```
def g(l):  
    if len(l)>1:  
        aux = l[:len(l)//2]  
        g(aux)  
        for el in l: print(el)  
        g(aux)  
    else: print(len(l))
```

4 Folosind metoda Divide et impera scrieți o funcție pură care primește o lista de numere și returnează o altă listă în care numerele sunt dublate. Ex. [1,2,3,4]->[2,4,6,8]. Datele se împart în 2 părți egale la fiecare pas (2p).

5 Găsiți sublista cea mai lungă cu numere pare crescătoare. Folosiți programare dinamică, se cere recurența și implementare iterativă în Python. Ex: Pentru lista [22, 2, 11,10, 4, 7, 8] soluția este 2, 4, 8 (2p)

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

Fiecare pagină, în colțul din dreapta sus, să conțină: nume prenume, grupă, 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.