Durata: 2 ore

1 a) **(0,5p)** Implementaţi o funcţie recursivă **suma_cifre** care primeşte ca parametru un număr natural **n** şi returnează suma cifrelor sale

b) **(1p)** Implementaţi o funcţie **numar_suma** care primeşte un număr variabil de numere naturale şi un număr **s** şi returnează câte dintre numerele întregi primite ca parametru au suma cifrelor cel puţin **s** (folosind funcţia de la a)). Apelaţi funcţia **numar_suma** pentru numerele 15, 19,178 şi 2856 şi cu valoarea parametrului **s** egală cu 10 (parametrul actual corespunzător lui s la apel este 10).

2) a) **(0.75 p)** Înlocuiţi punctele de suspensie din instrucţiunea *palindrom* = [...] cu o expresie (secvenţă de iniţializare/comprehensiune) astfel încât lista să fie iniţializată numerele palindrom de 3 cifre (**n** este palindrom dacă este egal cu inversul său).

b) **(0.75 p)** Ce afişează următoarea secvenţă de cod? Justificaţi.

```
def f(lista):
        a = b
        lista.append(a)
        lista.sort()
        lista = lista[:-1]
        lista.append(a+b)
        print(lista)
l = [4,2,5,10]
a = 1
b = 3
f(l)
print(l)
print(a)
print(b)
```

3 probleme de tehnici de programare din care alegeţi 2, fiecare având **3 puncte**:

Rezolvările subiectelor de tehnici de programare trebuie să conţină:

- o scurtă descriere a algoritmului şi o argumentare a faptului că acesta se încadrează într-o **anumită tehnică de programare** (deci şi o scurtă descriere a tehncii)
- implementarea Python în care se va preciza, pe scurt, sub forma unor comentarii, semnificaţia variabilelor utilizate.

3) Se dau un vector de numere întregi cu elementele ordonate crescător (pe o linie separate prin spaţiu) şi un număr **x**. Scrieţi un algoritm eficient pentru a determina **două** poziţii pe care apare **x** în vectorul **v** folosind un algoritm de tip *Divide et Impera*. Dacă **x** nu este element în vector se va afişa mesajul „nu este element”.

4) Se citeşte un număr natural n. Să se genereze folosind metoda *Backtracking* şi să se afişeze pe ecran toate numerele cu n cifre care au cifrele ordonate crescător (nestrict).

5) Se dă o mulţime de n intervale închise (se dă n pe o linie, apoi pe câte o linie extremitatea iniţială şi finală a fiecărui interval). Să se determine folosind metoda *Greedy* o submulţime de cardinal maxim de intervale disjuncte două câte două din mulţimea dată.