UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DOMENIUL DE STUDII INFORMATICĂ – ANUL I

EXAMEN LA DISCIPLINA "PROGRAMAREA ALGORITMILOR"
– model de subiect –

Subiectul I (3 p.)

a) Scrieți o funcție *divizori* care primește un număr variabil de parametri numere naturale și returnează pentru fiecare număr primit ca parametru lista divizorilor săi primi sub forma unui dicționar cu perechi de forma *număr*: *lista divizorilor*. De exemplu, pentru apelul *divizori*(50, 21) funcția trebuie să furnizeze dicționarul {50: [2,5], 21: [3,7]}. **(1.5 p.)**

b) Înlocuiți punctele de suspensie din instrucțiunea *litere_10* = [...] cu o expresie astfel încât lista să fie inițializată cu primele 10 litere mici din alfabetul englez. **(0.5 p.)**

c) Considerăm o funcție recursivă a cărei complexitate este dată de următoarea relație de recurență:

$T(1) = T(2) = 1$

$T(n) = T(n/3) + 2$, pentru $n \geq 1$

Determinați complexitatea funcției respective. (**1 p.**)

Subiectul 2 – metoda Greedy (3 p.)

Complexitatea maximă a soluției: $O(n \log_2 n)$

Considerăm n spectacole $S_1, S_2, ..., S_n$ pentru care cunoaștem intervalele lor de desfășurare $[s_1, f_1), ..., [s_n, f_n)$, toate dintr-o singură zi. Având la dispoziție o singură sală, în care putem să planificăm un singur spectacol la un moment dat, să se determine numărul maxim de spectacole care pot fi planificate fără suprapuneri. Un spectacol S_j poate fi programat după spectacolul S_i dacă $s_j \geq f_i$. Justificați corectitudinea programului și complexitatea sa.

Subiectul 3 – metoda Programării Dinamice (3 p.)

Complexitatea maximă a soluției: $O(n^2)$

Să se determine un subșir crescător de lungime maximă al unui șir t format din n numere întregi.

Subiectul 4 – metoda Backtracking (3 p.)

Să se afișeze toate permutările mulțimii $A = \{1,2, ..., n\}$, unde n este un număr natural nenul .

NOTĂ:

1. Toate subiectele se vor rezolva folosind limbajul Python.
2. Subiectul 1 este obligatoriu, iar dintre subiectele 2, 3 și 4 se vor rezolva CEL MULT DOUĂ, la alegere.
3. Citirea datelor de intrare se va realiza de la tastatură, iar rezultatele vor fi afișate pe ecran.
4. Se garantează faptul că datele de intrare sunt corecte.

5. Operaţiile de sortare se vor efectua folosind funcţii sau metode predefinite din limbajul Python.
6. Rezolvările subiectelor alese dintre subiectele 2, 3 şi 4 trebuie să conţină:
   - o scurtă descriere a algoritmului şi o argumentare a faptului că acesta se încadrează într-o anumită tehnică de programare;
   - în cazul problemelor rezolvate folosind metoda Greedy sau metoda programării dinamice se va argumenta corectitudinea criteriului de selecţie sau a relaţiilor de calcul;
   - în cazul subiectelor unde se precizează complexitatea maximă pe care trebuie să o aibă soluţia, se va argumenta complexitatea soluţiei propuse şi vor primi punctaj maxim doar soluţiile corecte care se încadrează în complexitatea cerută;
   - în fiecare program Python se va preciza, pe scurt, sub forma unor comentarii, semnificaţia variabilelor utilizate.
7. Pentru subiectele 1 nu contează complexităţile soluţiilor propuse.
8. Rezolvările corecte care nu respectă restricţiile indicate vor primi punctaje parţiale.
9. Se acordă 1 punct din oficiu.