

Subiectul III (30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Utilizând metoda backtracking se generează toate permutările mulțimii $\{1, 2, 3, 4\}$. Dacă primele trei permutări generate sunt, în această ordine: 1234, 1243, 1324 precizați care este permutarea generată imediat după 3412. **(4p.)**
- a. 3214 b. 3413 c. 4123 d. 3421

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Subprogramul `f` are definiția alăturată. Ce valoare are `f(7)`? Dar `f(100)`? **(6p.)**
- ```
int f(int x)
{
 if(x%6==0) return x;
 else return f(x-1);
}
```
3. Scrieți definiția completă a unui subprogram `P`, cu trei parametri, care primește prin intermediul primului parametru, `a`, un tablou unidimensional de cel mult 100 de numere întregi, cu cel mult 4 cifre fiecare, prin intermediul celui de-al doilea parametru, `n`, numărul efectiv de elemente ale tabloului, iar prin parametrul `k`, un număr natural ( $k < n$ ) și returnează cea mai mare sumă cu `k` termeni care se poate obține adunând `k` elemente ale tabloului.
- Exemplu:** dacă `n=6` și `k=4`, iar șirul este format din elementele (5, 2, 5, 4, 1, 3), atunci la apel se va returna 17. **(10p.)**
4. Fișierul text `numere.txt` conține pe prima linie un număr natural `n` ( $0 < n < 100000$ ) iar pe a doua linie, separate prin câte un spațiu, `n` numere naturale formate din cel mult două cifre fiecare.
- a) Scrieți un program C/C++ care determină în mod eficient, din punct de vedere al timpului de executare, numerele ce apar o singură dată în a doua linie a fișierului. Aceste numere vor fi afișate pe ecran în ordine crescătoare, separate prin câte un spațiu.
- Exemplu:** dacă fișierul `numere.txt` are următorul conținut:
- ```
7
3 5 2 1 5 23 1
```
- atunci pe ecran se va afișa: 2 3 23. **(6p.)**
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**