

Subiectul III (30 de puncte)

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

1. Utilizând metoda backtracking se generează numerele formate din câte 3 cifre distincte din mulțimea $\{1, 3, 5, 7\}$. Dacă primele trei numere generate sunt, în această ordine: 135, 137, 153 care este cel de-al patrulea număr generat? (4p.)
- a. 315 b. 173 c. 157 d. 357

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

2. Subprogramul `f` are definiția alăturată. Ce valoare are `f(3)`? Dar `f(10)`? (6p.)
- | | |
|--------------|-----------------------|
| int f(int x) | {if(x==0)return 0; |
| | else return f(x-1)+2; |
| | } |
3. Scrieți definiția completă a unui subprogram `P` cu doi parametri, care primește prin intermediul primului parametru, `n`, un număr natural nenul ($1 \leq n \leq 100$) și prin intermediul celui de-al doilea parametru, `a`, un tablou unidimensional cu elementele numerotate de la 1 la `n`, numere întregi, de cel mult 4 cifre fiecare, și returnează suma tuturor numerelor impare aflate pe poziții pare în tablou.
Exemplu: dacă `n=6`, iar șirul este format din elementele (3, 2, 7, 1, 4, 3), atunci la apel se va returna 4. (10p.)
4. Fișierul text `numere.txt` conține pe prima linie un număr natural `n` ($0 < n < 100000$), iar pe a doua linie `n` cifre, separate prin câte un spațiu.
- a)** Scrieți un program C/C++ care determină în mod eficient, din punct de vedere al timpului de executare, cea mai mare cifră dintre cele situate pe a doua linie a fișierului `numere.txt` și numărul de apariții ale acesteia. Valorile determinate se vor afișa pe ecran, separate printr-un spațiu.
Exemplu: dacă fișierul `numere.txt` are următorul conținut:
- ```
11
2 5 3 1 5 8 9 2 7 3 4
```
- atunci pe ecran se va afișa: 9 1. (6p.)
- b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)