

Subiectul III (30 de puncte)

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

1. Se generează, prin metoda backtracking, toate partițiile mulțimii $A = \{1, 2, 3\}$ obținându-se următoarele soluții: $\{1\}\{2\}\{3\}$; $\{1\}\{2, 3\}$; $\{1, 3\}\{2\}$; $\{1, 2\}\{3\}$; $\{1, 2, 3\}$. Se observă că dintre acestea, prima soluție e alcătuită din exact trei submulțimi. Dacă se folosește aceeași metodă pentru a genera partițiile mulțimii $\{1, 2, 3, 4\}$ stabiliți câte dintre soluțiile generate vor fi alcătuite din exact trei submulțimi. **(4p.)**
- a. 3 b. 12 c. 6 d. 5

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

2. Se consideră tabloul unidimensional **a** definit global, ce memorează elementele $a_1=1$, $a_2=2$, $a_3=0$ și subprogramul **f**, definit alăturat. Ce valoare are $f(2,1)$? Dar $f(3,3)$? **(6p.)**
- ```
int f(int b,int i)
{if(i>=1)
 return f(b,i-1)*b+a[i];
else return 0;
}
```
3. Subprogramul **verif** primește prin singurul său parametru, **x**, un număr natural nenul cu cel mult 9 cifre și returnează valoarea 1 dacă numărul conține cel puțin o secvență de 3 cifre impare alăturate și 0 în caz contrar.  
**Exemplu:** dacă  $x=7325972$  se va returna valoarea 1.  
**a)** Scrieți definiția completă a subprogramului **verif**. **(6p.)**  
**b)** Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** cu exact 6 cifre și, folosind apeluri utile ale subprogramului **verif**, verifică dacă **n** are primele trei cifre impare. Programul afișează pe ecran mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.  
**Exemple:** dacă se citește  $n=132567$  se va afișa mesajul **Nu**, iar dacă  $n=979243$ , se va afișa mesajul **Da**. **(4p.)**
4. Pentru un șir de numere naturale, numim "nod" al șirului un termen din șir care are doi vecini, termenul precedent și termenul următor din șir, și valoarea termenului respectiv este strict mai mică decât suma valorilor celor doi vecini ai săi.  
**a)** Fișierul text **date.in** conține un șir de cel puțin două și cel mult 10000 de numere naturale având maximum 6 cifre fiecare, numere separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și afișează numărul de "noduri" ale șirului citit, folosind un algoritm eficient din punctul de vedere al memoriei utilizate. **(6p.)**  
**Exemplu:** dacă fișierul **date.in** are următorul conținut:  
51   20   100   43   43   618   5000   31   2020   114   116   4  
atunci pe ecran se afișează 7 (cele șapte numere subliniate reprezintă "noduri" ai șirului)  
**b)** Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**