

Subiectul III (30 de puncte)

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

1. Se generează, utilizând metoda backtracking, toate modalitățile de așezare a numerelor naturale de la 1 la 5, astfel încât oricare două numere de aceeași paritate să nu se afle pe poziții alăturate. Dacă primele 3 soluții generate sunt, în ordine: (1,2,3,4,5), (1,2,5,4,3) și (1,4,3,5,2) care este prima soluție generată ce începe cu 3? **(4p.)**
- a. (3, 2, 1, 4, 5) b. (3, 2, 5, 4, 1) c. (3, 4, 1, 2, 5) d. (3, 4, 5, 2, 1)

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

2. Subprogramul recursiv alăturat este definit incomplet. Care este expresia cu care se pot înlocui punctele de suspensie astfel încât subprogramul să returneze cel mai mare divizor comun al numerelor primite prin intermediul parametrilor **a** și **b**. **(6p.)**
- ```
int f(int a,int b)
{ if (...)
 return a;
else
 if (a>b) return f(a-b,b);
 else return f(a,b-a);
}
```
3. Subprogramul **sfx** primește prin singurul său parametru, **x**, un număr natural din intervalul [100,200000000] și returnează valoarea 1 dacă ultimele trei cifre ale numărului sunt în ordine strict descrescătoare sau valoarea 0 în caz contrar.  
**Exemplu:** dacă **x=24973** se va returna valoarea 1.
- a) Scrieți definiția completă a subprogramului **sfx**. **(5p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural **n** format din exact 6 cifre și verifică, utilizând apeluri ale subprogramului **sfx**, dacă acest număr are toate cifrele în ordine strict descrescătoare. Programul va afișa mesajul **Da** în caz afirmativ și mesajul **Nu** în caz contrar.  
**Exemplu:** dacă **n=756543** se va afișa **Nu**, iar dacă **n=976532** se va afișa **Da**. **(5p.)**
4. Pentru un șir de numere naturale, numim "pol" al șirului un termen din șir care are doi vecini, termenul precedent și termenul următor din șir, și valoarea termenului respectiv este strict mai mare decât valoarea fiecăruia dintre cei doi vecini ai săi.
- a) Fișierul **text.dat** conține un șir de cel puțin două și 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 "poli" ai șirului citit, folosind un algoritm eficient din punctul de vedere al memoriei utilizate. **(6p.)**  
**Exemplu:** dacă fișierul **dat** are următorul conținut:
- ```
51 20 100 43 43 618 5000 31 2020 114 116 4
```
- atunci pe ecran se afișează 4 (cele patru numere subliniate reprezintă "poli" ai șirului)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**