

Subiectul III (30 de puncte)

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

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

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

2. Se consideră funcția f , definită alăturat.
Ce se afișează ca urmare a executării secvenței de mai jos în care variabilele a și b sunt de tip `int`?
`a=4; b=18;`
`printf("%d",f(a,b));/ cout<<f(a,b);`
`printf("%d %d",a,b);/ cout<<a<<b;` (6p.)
- ```
int f(int &a, int &b)
{ while (a !=b)
 if (a>b) a=a-b;
 else b=b-a;
 return a;}
```
3. Subprogramul `sfx` primește prin singurul său parametru,  $x$ , un număr natural din intervalul  $[100, 2000000000]$  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 date.in` conține un șir de 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 "poli" ai 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 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.)