

Subiectul III (30 de puncte)

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

- | | |
|---|---|
| <p>1. Funcția F are definiția alăturată. Ce valoare are F(5)? (4p.)</p> <p>a. 5 b. 10 c. 15 d. 6</p> | <pre>int F(int x) {if(x!=0) return x+F(x-1); else return x;}</pre> |
|---|---|

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

2. Un algoritm generează în ordine descrescătoare, toate numerele de **n** cifre ($n < 9$), cu cifrele în ordine strict crescătoare, care nu au două cifre pare alăturate. Dacă pentru **n=5**, primele cinci soluții generate sunt **56789**, **45789**, **45679**, **45678**, **36789**, precizați care sunt următoarele **trei** soluții generate, în ordinea obținerii lor. (6p.)
3. Subprogramul **nule** are doi parametri: **a**, prin care primește un tablou unidimensional cu maximum 100 de numere întregi, cu cel mult 4 cifre fiecare și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile **nule** să se afle la sfârșitul tabloului. Ordinea în cadrul secvenței de elemente **nule** poate fi oricare. Tabloul modificat este furnizat tot prin parametrul **a**.
Exemplu: dacă **n=6**, **a=(12,0,0,-3,-8,0)**, după apel, acesta ar putea fi:
a=(12,-3,-8,0,0,0). (10p.)
- a) Scrieți definiția completă a subprogramului **nule**. (10p.)
- b) Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** ($2 \leq n \leq 100$) și apoi un șir de **n** numere întregi cu cel mult 4 cifre fiecare și care, folosind apeluri utile ale subprogramului **nule**, verifică dacă șirul conține cel puțin cinci valori **nule**, nu neapărat distincte. În caz afirmativ programul afișează cinci dintre acestea, iar altfel mesajul **NU EXISTA**.
Exemplu: pentru **n=11** și pentru șirul **6,0,0,16,10,0, 8,-18,0,0,20** se poate afișa pe ecran soluția **6 16 10 8 -18**. (6p.)
4. În fișierul **numere.txt** sunt memorate cel puțin 4 și cel mult 90 de numere întregi cu cel mult patru cifre fiecare, separate prin câte un spațiu. Scrieți un program **C/C++** care afișează pe ecran **patru** numere aflate pe poziții consecutive în fișier, care sunt în ordine strict descrescătoare. Dacă există mai multe astfel de secvențe programul afișează una dintre acestea, iar dacă în fișier nu există astfel de secvențe se afișează mesajul **NU EXISTA**.
Exemplu: dacă fișierul **numere.txt** conține, în această ordine, numerele **60 120 15 5 2 45 25** se vor afișa numerele **120 15 5 2**. (6p.)