## Subiectul III (30 de puncte)

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

**1.** La examenul de bacalaureat, un elev primeşte un test format dintr-un subiect de tip **I**, unul de tip **II** şi unul de tip **III**. Stiind că pentru fiecare tip de subiect sunt elaborate exact **100** de variante, algoritmul de generare a tuturor posibilităților de a forma un test este similar cu algoritmul de generare a: **(4p.)**

**a.** elementelor produsului cartezian
**b.** aranjamentelor
**c.** permutărilor
**d.** submulțimilor

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

**2.** Se consideră subprogramul **f**, definit alăturat. Ce se afişează la apelul **f(4);**? **(6p.)**

```
void f(int n)
{
 cout<<"*";  |  printf("*");
 if(n>2)
  {
   f(n-1);
   cout<<"#"; |  printf("#");
  }
}
```

**3.** Scrieți definiția completă a subprogramului **numar**, cu doi parametri, care primeşte prin intermediul parametrului **n** un număr natural format din cel mult **9** cifre, iar prin intermediul parametrului **c** o cifră nenulă; subprogramul returnează numărul obținut prin înlocuirea fiecărei apariții a cifrei **c** în numărul **n** cu partea întreagă a valorii **(c+1)/2**. Dacă **c** nu apare în **n**, subprogramul returnează valoarea **n**.
**Exemplu:** pentru **n=1525735**, **c=5** valoarea returnată va fi **1323733**. **(10p.)**

**4.** Fişierul text **bac.txt** conține cel puțin două şi cel mult **1000** de numere naturale, **distincte**. Numerele sunt separate prin câte un spațiu şi fiecare dintre ele are cel mult **9** cifre.

**a)** Scrieți un program **C/C++** care determină cele mai mici două numere din fişier, utilizând un algoritm eficient din punct de vedere al timpului de executare şi al spațiului de memorie utilizat. Cele două numere vor fi afişate pe ecran, în ordine **crescătoare**, separate printr-un spațiu. **(6p.)**
**Exemplu:** dacă fişierul conține numerele: **512 6 860 3 12 300** se va afişa: **3 6**

**b)** Descrieți succint, în limbaj natural, algoritmul utilizat, justificând eficiența acestuia. **(4p.)**