

Subiectul III (30 de puncte) - Varianta 091

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

1. Pentru a planifica în orarul unei școli, la clasa a XII-a, 4 ore de informatică în zile lucrătoare diferite din săptămână, câte o singură oră pe zi, se poate utiliza un algoritm de generare echivalent cu algoritmul de: **(4p.)**
- a. generare a permutărilor de 4 elemente b. generare a aranjamentelor de 4 elemente luate câte 5
- c. generare a aranjamentelor de 5 elemente luate câte 4 d. generare a combinațiilor de 5 elemente luate câte 4

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

2. Subprogramul recursiv alăturat este definit incomplet. Scrieți expresia care poate înlocui punctele de suspensie astfel încât, în urma apelului, **f(12)** să se afișeze șirul de valori: **12 6 3 1 1 3 6 12.** **(6p.)**
- ```
void f(int i)
{ if (. . .)
 { printf("%d ",i); | cout<<i<<' ';
 f(i/2);
 printf("%d ",i); | cout<<i<<' ';
 }
}
```
3. Subprogramul **par** primește prin singurul său parametru, **n**, un număr natural nenul cu cel mult 4 cifre și returnează valoarea **1** dacă **n** conține cel puțin o cifră pară, sau returnează valoarea **0** în caz contrar.  
**Exemplu:** pentru **n=723** subprogramul va returna valoarea **1**.
- a) Scrieți numai antetul subprogramului **par**. **(2p.)**
- b) Scrieți un program **C/C++** care citește de la tastatură un număr natural nenul **n** cu cel mult 9 cifre, apoi un șir de **n** numere naturale, cu exact 8 cifre fiecare, și afișează pe ecran, numărul de valori din șirul citit care au cel puțin o cifră pară printre primele 4 poziții, considerate de la stânga spre dreapta. Se vor utiliza apeluri utile ale subprogramului **par**.  
**Exemplu:** dacă **n=3**, iar șirul citit este **12386406, 37152073, 73543157** se va afișa **2** (numerele **12386406** și **73543157** respectă condiția cerută). **(8p.)**
4. Fișierul **numere.in** conține cel mult **5000** de numere reale, câte unul pe fiecare linie. Se cere să se scrie un program care să citească toate numerele din fișier și să afișeze pe ecran numărul de ordine al primei, respectiv al ultimei linii pe care se află cel mai mare număr din fișier. Cele două numere vor fi separate printr-un spațiu. Alegeți o metodă de rezolvare eficientă din punct de vedere al spațiului de memorare și al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul alăturat, pe ecran se vor afișa numerele **2 6.**
- |     |
|-----|
| 3.5 |
| 7   |
| -4  |
| 7   |
| 2   |
| 7   |
| 6.3 |
| 5   |
- a) Descrieți succint, în limbaj natural, metoda de rezolvare aleasă, explicând în ce constă eficiența ei. **(4p.)**
- b) Scrieți programul **C/C++** corespunzător metodei descrise. **(6p.)**