

Subiectul III (30 de puncte) - Varianta 074

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

1. Prin metoda backtracking se generează toate anagramele (cuvintele obținute prin permutarea literelor) unui cuvânt dat. Știind că se aplică această metodă pentru cuvântul **pescar**, precizați câte cuvinte se vor genera astfel încât prima și ultima literă din fiecare cuvânt generat să fie **vocală** (sunt considerate vocale caracterele **a, e, i, o, u**)? **(4p.)**
- a. 96 b. 24 c. 48 d. 12

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

2. Se consideră funcția **Suma**, definită alăturat. Ce valoare are **Suma(8)**? Dar **Suma(11)**? **(6p.)**
- ```
int Suma(int x)
{ if(x == 1) return 0;
 if(x%2==0) return Suma(x-1)+(x-1)*x;
 return Suma(x-1)-(x-1)*x;
}
```
3. Scrieți definiția completă a subprogramului **Cifra**, cu doi parametri, **n** și **x**, care primește prin intermediul parametrului **n** un număr natural de cel mult nouă cifre și furnizează prin parametrul **x** cea mai mare cifră a numărului transmis prin parametrul **n**. **(10p.)**
4. a) Se numește “număr mare” un număr care are mai mult de nouă cifre. Scrieți un program **C/C++** care citește din fișierul text **NUMERE.IN** un număr natural **n** (**10 < n < 1000**) apoi de pe următoarea linie **n** numere naturale cu cel mult nouă cifre fiecare, dintre care cel puțin unul nenul, și folosind apeluri ale funcției **Cifra** definită la punctul 3 construiește în mod eficient din punct de vedere al memoriei folosite, cel mai mic “număr mare” cu exact **n** cifre. Fiecare dintre cifrele numărului mare reprezintă cifra de valoare maximă a unui număr citit din fișier și nu există două cifre în “numărul mare” care să aparțină aceluiași număr citit. Scrieți în fișierul text **NUMERE.OUT** “numărul mare” obținut astfel. **(6p.)**
- b) Descrieți succint în limbaj natural metoda de rezolvare folosită explicând în ce constă eficiența ei (3 – 4 rânduri). **(4p.)**

**Exemplu:**

**NUMERE.IN**

**10**

**725 3695 423 0 7895 0 100 101 870 568**

**NUMERE.OUT**

**1001478899**