

Subiectul III (30 de puncte)

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

1. Se utilizează metoda backtracking pentru a genera în **ordine lexicografică** toate cuvintele care conțin toate literele din mulțimea **{a,m,i,c}**, astfel încât fiecare literă să apară exact o dată într-un cuvânt. Câte soluții sunt generate după cuvântul **amic** și înainte de cuvântul **cami**? **(4p.)**
- a. 6 b. 4 c. 1 d. 3

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

2. Se consideră subprogramul **f**, definit alăturat. Ce valoare are **f(10)**?
Dar **f(8261)**? **(6p.)**
- ```
int f(int a)
{
 if(a<10)
 return 7;
 return f(a/100)*10+8;
}
```
3. Fișierul text **bac.txt** conține, pe o singură linie, cel puțin 2 și cel mult 100 de numere naturale nenule distincte de cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și scrie pe ecran ultima cifră a produsului celor mai mari două numere dintre cele citite.  
**Exemplu:** dacă fișierul **bac.txt** conține numerele:  
**1017 48 312 5742 162**  
atunci se va afișa: **4** (ultima cifră a produsului numerelor **1017, 5742**) **(10p.)**
4. Se consideră subprogramul **divizor**, cu doi parametri, **a** și **d**, care:
- primește prin intermediul parametrului **a** un număr natural nenul de cel mult 4 cifre, strict mai mare ca 1;
  - furnizează prin intermediul parametrului **d**, cel mai mic divizor al lui **a** strict mai mare decât 1.
- a) Scrieți numai antetul subprogramului **divizor**. **(4p.)**
- b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură un număr natural **x** (**x>1**), și, prin apeluri utile ale subprogramului **divizor**, verifică dacă **x** este număr prim. Programul va afișa pe ecran, în caz afirmativ, mesajul **DA**, iar în caz contrar mesajul **NU**. **(6p.)**