

**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 de câte patru litere din mulțimea {**d, a, n, s**}, astfel încât în niciun cuvânt să nu existe două litere alăturate identice. Știind că primele trei cuvinte generate sunt, în ordine, **ada**d, **ada**n și **ada**s, care va fi ultimul cuvânt obținut? **(4p.)**
- a. **snns**                      b. **nsns**                      c. **snsn**                      d. **dans**

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

2. Se consideră subprogramul **f**, definit alăturat. Ce se va afișa la apelul **f(38);**? **(6p.)**
- ```
void f(int x)
{
    if(x) {
        if(x%3==0){
            cout<<3; | printf("3");
            f(x/3);
        }
        else{
            f(x/3);
            cout<<x%3; | printf("%d",x%3);
        }
    }
}
```

3. Fișierul text **INTRARE.TXT** conține, pe o singură linie, cel mult 100 de numere naturale nenule de cel mult patru cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișier și scrie în fișierul text **IESIRE.TXT**, în ordine crescătoare, toate valorile distincte obținute ca sumă de două elemente distincte aflate pe prima linie a fișierului **INTRARE.TXT**.

**Exemplu:**

| <b>INTRARE.TXT</b> | <b>IESIRE.TXT</b> |
|--------------------|-------------------|
| 1 4 3 2            | 3 4 5 6 7         |

**(10p.)**
4. Se consideră subprogramul **multiplu**, cu doi parametri, care:
- primește prin intermediul parametrilor **a** și **k** două numere întregi de cel mult 4 cifre;
  - returnează cel mai mic multiplu al lui **k** mai mare sau egal cu **a**.
- a) Scrieți numai antetul subprogramului **multiplu**. **(4p.)**
- b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură trei numere naturale nenule **x**, **y**, **z**, de cel mult 4 cifre fiecare, (**x**≤**y**), și care, prin apeluri utile ale subprogramului **multiplu**, verifică dacă intervalul [**x**,**y**] conține cel puțin un multiplu al lui **z**. Programul va afișa pe ecran, în caz afirmativ, mesajul **DA**, iar în caz contrar mesajul **NU**. **(6p.)**