



**SUBIECTUL al II-lea**

**(30 de puncte)**

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. În declararea alăturată, câmpurile **a** și **b** ale înregistrării reprezintă lungimea și lățimea unui dreptunghi. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă dreptunghiul corespunzător variabilei **d** este pătrat. **(4p.)**
- |                                 |   |
|---------------------------------|---|
| <b>a.</b> <code>a.d==b.d</code> | <b>b.</b> <code>a-&gt;dreptunghi==b-&gt;dreptunghi</code> |
| <b>c.</b> <code>d.a==d.b</code> | <b>d.</b> <code>dreptunghi.a==dreptunghi.b</code>         |
2. Se consideră un graf neorientat cu 7 noduri și două componente conexe. Numărul de muchii ale grafului **NU** poate fi: **(4p.)**
- |             |              |              |              |
|-------------|--------------|--------------|--------------|
| <b>a.</b> 5 | <b>b.</b> 11 | <b>c.</b> 15 | <b>d.</b> 21 |
|-------------|--------------|--------------|--------------|

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

3. Variabilele **n** și **k** sunt de tip întreg ( $1 \leq k \leq n$ , **n** par), iar variabila **a** memorează elementele unui tablou bidimensional cu **n** linii și **n** coloane, numerotate de la 1 la **n**.  
Scrieți o secvență de program C/C++ care să afișeze pe ecran cea mai mică dintre cele două valori aflate pe linia **k** și pe diagonala principală a tabloului, respectiv pe linia **k** și pe diagonala secundară a tabloului. **(6p.)**
4. Într-un arbore cu rădăcină considerăm că un nod se află pe nivelul **x** dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea **x**.  
Arborele cu 8 noduri, cu etichete de la 1 la 8, are muchiile [1,3], [1,7], [2,3], [2,5], [4,5], [5,6], [5,8]. Scrieți etichetele tuturor nodurilor care pot fi alese drept rădăcină astfel încât, în fiecare dintre aceste cazuri, numărul de niveluri ale arborelui să fie minim. **(6p.)**
5. Se consideră un text cu cel mult 70 de caractere (litere mici ale alfabetului englez și spații), în care cuvintele sunt separate prin unul sau mai multe spații. Înaintea primului cuvânt și după ultimul cuvânt nu există spații.  
Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat mai sus și afișează pe ecran numărul de cuvinte în care apare litera **a**.  
**Exemplu:** pentru textul  
`voi sustine examenul la informatica`  
se afișează 3. **(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Se consideră subprogramele **f1** și **f2**, definite mai jos, în care **a** este nenul.

```
int f1(int a, int b)
{
    return (b/a)*a;
}
```

```
int f2 (int a, int b)
{ if(b%a==0)
    return b;
  return f2(a, b-1);
}
```

Identificați subprogramul care, la apel, pentru parametrii **a=3** și **b=10**, returnează cel mai mare multiplu al lui **a** mai mic sau egal cu **b**. **(4p.)**

- a. atât **f1**, cât și **f2**    b. numai **f1**    c. numai **f2**    d. nici **f1**, nici **f2**

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

2. Utilizând metoda backtracking se generează toate numerele cu cifre distincte și nenule, numere care au suma cifrelor egală cu 10. Primele patru soluții generate sunt, în această ordine: 1234, 1243, 127, 1324. Scrieți cea de a cincea și cea de a șasea soluție, în ordinea generării acestora. **(6p.)**

3. Se consideră subprogramul **num**, cu doi parametri:

- **n**, prin care primește o valoare naturală  $2 < n < 50$ ;
- **v**, prin care primește un tablou unidimensional cu **n** elemente, numere întregi cu cel mult 4 cifre.

Subprogramul înlocuiește cu 0 fiecare valoare mai mică sau egală cu prima valoare din tablou. Tabloul modificat este furnizat tot prin parametrul **v**.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **n=7** și **v=(4, 5, 0, 9, 3, 4, -2)**, atunci după apel

**v=(0, 5, 0, 9, 0, 0, 0)**.

**(10p.)**

4. Fișierul **numere.in** memorează cel mult 1000000 de numere naturale cu cel mult nouă cifre. Numerele sunt ordonate strict crescător și separate prin câte un spațiu.

Se consideră șirul 1, 4, 7 . . . . definit astfel:  $f_1=1$ ,  $f_2=4$  și  $f_n=2 \cdot f_{n-1}-f_{n-2}$ , pentru  $n>2$ .

Se cere să se afișeze pe ecran numerele din fișier care sunt termeni ai șirului. Numerele sunt afișate în ordine strict crescătoare, separate prin câte un spațiu. Dacă nu există astfel de numere se afișează pe ecran mesajul **Nu exista**.

Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

**Exemplu:** dacă fișierul **numere.in** conține numerele 1 2 5 7 17 30, se afișează pe ecran numerele 1 7.

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

- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**