



**SUBIECTUL al II-lea**

**(30 de puncte)**

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

1. Variabila **a** memorează elementele unui tablou bidimensional cu 100 de linii și 100 de coloane, numerotate de la 1 la 100. Un element aflat pe diagonala secundară a tabloului poate fi accesat prin: **(4p.)**
- a. **a[1][3]**                      b. **a[15][15]**                      c. **a[16][24]**                      d. **a[42][59]**
2. 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**. Pe nivelul 0 se află un singur nod (rădăcina).  
Se consideră un arbore cu rădăcină, cu patru niveluri. Toate nodurile de pe același nivel (cu excepția ultimului nivel) au un număr egal (nenul) de descendenți direcți („fii”) și nu există două niveluri cu același număr de noduri. Numărul minim de noduri de pe nivelul 3 este: **(4p.)**
- a. **6**                                      b. **8**                                      c. **9**                                      d. **12**

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

3. În declararea alăturată, variabilele **f** și **fs** memorează în câmpurile **a** și **b** numărătorul, respectiv numitorul câte unei fracții. Scrieți o secvență de instrucțiuni C/C++ care să memoreze în variabila **fs** fracția obținută prin însumarea fracției memorate în variabila **f** și a fracției  $\frac{2}{3}$ . **(6p.)**
- ```
struct fractie {  
    int a,b;  
} f,fs;
```
4. Scrieți toate ciclurile elementare distincte ale unui graf neorientat cu 6 noduri, numerotate de la 1 la 6, și muchiile [1,2], [1,3], [1,5], [3,4], [3,5], [4,5], [5,6].  
Două cicluri se consideră distincte dacă ele diferă prin cel puțin o muchie. **(6p.)**
5. Se consideră un text cu cel mult 100 de caractere (litere mici ale alfabetului englez și spații), în care cuvintele sunt separate prin câte un spațiu. 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 determină transformarea acestuia în memorie prin înlocuirea fiecărui cuvânt format din trei litere cu simbolul \*. Programul afișează pe ecran textul obținut.  
**Exemplu:** pentru textul  
bun este izvorul ce are apa rece  
se afișează  
\* este izvorul ce \* \* rece **(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.

```
int f1(int n)
{ return n*(n+1)/2;
}
```

```
int f2 (int n)
{ if(n>0)
    return n+f2(n-1);
    return 0;
}
```

Identificați subprogramul care, la apel, pentru parametrul **n=10**, returnează suma primelor 10 numere naturale. **(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 submulțimile cu cel mult patru elemente din mulțimea {3, 5, 6, 7, 8}. Primele șase soluții generate sunt, în această ordine: {3}, {3, 5}, {3, 5, 6}, {3, 5, 6, 7}, {3, 5, 6, 8}, {3, 5, 7}. Scrieți cea de a șaptea și cea de a opta soluție, în ordinea generării acestora. **(6p.)**

3. Se consideră subprogramul **sub**, cu trei parametri:

- **n**, prin care primește un număr natural ( $2 < n < 50$ );
- **v**, prin care primește un tablou unidimensional cu **n** elemente, numere naturale cu cel mult 4 cifre;
- **x**, prin care primește un număr natural cu cel mult 4 cifre. Cel puțin unul dintre elementele tabloului are valoarea **x**.

Subprogramul modifică ordinea valorilor din tablou, astfel încât toate valorile egale cu **x** să ocupe primele poziții din **v**, iar celelalte valori să se regăsească în continuarea acestora. Tabloul modificat este furnizat tot prin parametrul **v**.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă **n=9**, **v**=(2, 1, 0, 1, 7, 0, 1, 4, 5) și **x=1**, atunci, după apel, o soluție posibilă este **v**=(1, 1, 1, 2, 0, 7, 0, 4, 5). **(10p.)**

4. Fișierul **bac.txt** conține un șir de cel puțin trei și cel mult 1000000 de numere naturale cu cel mult nouă cifre. Numerele din șir sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran, separate printr-un spațiu, două numere distincte, anume cel mai mic număr par cu două cifre și cel mai mare număr par cu două cifre care **NU** fac parte din șir.

Dacă nu există două astfel de valori, pe ecran se afișează mesajul **Nu exista**.

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

**Exemplu:** dacă fișierul **bac.txt** conține valorile

7 2 40 5 10 15 11 12 18 350

se afișează pe ecran numerele 14 98.

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

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