



**SUBIECTUL al II-lea**

**(30 de puncte)**

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

- Variabila **a** memorează elementele unui tablou bidimensional cu 100 de linii și 100 de coloane, numerotate de la 0 la 99. În limbajul C/C++ un element aflat pe linia 21 și coloana 50 a tabloului poate fi accesat prin: (4p.)
  - a (50, 21)**
  - a (50) (21)**
  - a [21; 50]**
  - a [21] [50]**
- 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 cinci 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 4 este: (4p.)
  - 4**
  - 9**
  - 16**
  - 18**

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

- Se consideră declararea alăturată. Scrieți o secvență de instrucțiuni în urma executării căreia câmpul **mii** al variabilei **x** să memoreze litera **M**, dacă numărul memorat în câmpul **v** al acesteia este strict mai mare decât 999, sau litera **A** altfel. (6p.)

```
struct numar  
{ int v;  
  char mii;  
} x;
```
- Scrieți toate ciclurile elementare distincte ale unui graf neorientat cu 6 noduri, numerotate de la 1 la 6, și muchiile [1, 2], [1, 4], [1, 5], [2, 3], [2, 4], [3, 4], [4, 6].  
Două cicluri se consideră distincte dacă ele diferă prin cel puțin o muchie. (6p.)
- Se consideră un text cu cel mult 100 de caractere (litere mici ale alfabetului englez și spații), în care cuvintele sunt separate prin unul sau mai multe 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 eliminarea unor vocale, astfel încât în niciun cuvânt din text să nu existe două vocale pe poziții alăturate. Programul afișează pe ecran textul obținut. Se consideră vocale literele a, e, i, o, u.  
**Exemplu:** pentru textul **o aia aceea e foarte blanda**  
se poate obține textul **o ace e forte blanda**  
iar pentru textul **ea e o oaie**  
se poate obține textul **e e o o** (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.

```
long f1 (int n)
{ if(n<10)
    return n*f1(n+1);
  return 10;
}
```

```
long f2 (int n)
{ if(n>1)
    return n*f2(n-1);
  return 1;
}
```

Identificați subprogramul care, la apel, pentru parametrul  $n=10$ , returnează produsul primelor 10 numere naturale nenule. (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. Alfabetul Morse este format doar din simbolurile linie (notat cu  $-$ ) și punct (notat cu  $\bullet$ ). Utilizând metoda backtracking se generează toate cuvintele scrise în alfabetul Morse, formate din câte 5 simboluri, care încep și se termină cu punct. Primele patru soluții generate sunt, în această ordine:

$\bullet - - - \bullet$      $\bullet - - \bullet \bullet$      $\bullet - \bullet - \bullet$      $\bullet - \bullet \bullet \bullet$

Scrieți cea de a cincea și cea de a șasea soluție, în ordinea generării acestora. (6p.)

3. Se consideră subprogramul `divizor`, cu doi parametri:

- $n$ , prin care primește un număr natural cu cel mult nouă cifre ( $n > 1$ );
- $d$ , prin care furnizează cel mai mare divizor prim al lui  $n$ .

Scrieți definiția completă a subprogramului.

**Exemplu:** pentru  $n=50$ , după apel  $d=5$ , iar pentru  $n=11$ , după apel  $d=11$ . (10p.)

4. Fișierul `bac.txt` conține pe prima linie un număr natural **par**  $n$  cu cel mult patru cifre, iar pe următoarea linie un șir de  $n$  numere naturale cu cel mult nouă cifre. Numerele din șir sunt în ordine descrescătoare și sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran cel mai mic număr din prima jumătate a șirului care să fie strict mai mare decât oricare număr din a doua jumătate a șirului. Dacă în fișier nu se află o astfel de valoare, pe ecran se afișează mesajul **Nu exista**.

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

**Exemplu:** dacă fișierul `bac.txt` are conținutul

30

16 8 7 7 ... 7 7 2 1

de 26 de ori

atunci pe ecran se afișează 8, iar dacă fișierul are conținutul

6

7 7 7 7 2 1

atunci pe ecran se afișează **Nu exista**.

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

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