

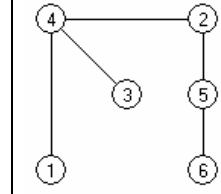
Subiectul II (30 de puncte)

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

1. Care este numărul maxim de noduri de grad 3 într-un graf neorientat cu 5 noduri? (4p.)

a. 4 b. 5 c. 3 d. 2

2. Într-un arbore cu rădăcină, nivelul unui nod este egal cu lungimea lanțului format din noduri distincte care unește rădăcina cu acel nod. Care dintre noduri trebuie ales ca rădăcină în arborele din figura alăturată astfel încât pe fiecare nivel să se găsească un număr impar de noduri? (6p.)



a. 2 b. 3 c. 6 d. 4

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

3. Care va fi șirul de caractere afișat pe ecran după executarea secvenței de program alăturate în care variabila **s** memorează un șir cu cel mult 4 caractere iar variabila **t** un caracter? (4p.)

```
char s[]="arac";  
t=s[1]; s[1]=s[3];  
s[3]='t';  
cout<<s; | printf("%s",s);
```

4. O listă liniară dublu înlănțuită, alocată dinamic, reține în câmpul **nr** al fiecărui element câte un număr natural, în câmpul **urm**, adresa elementului următor din listă sau **NULL** dacă nu există un element următor, iar în câmpul **prec**, adresa elementului precedent din listă sau **NULL** dacă nu există un element precedent. Variabilele **p** și **q** memorează adresa primului, respectiv ultimului element al listei.

```
n=0;  
while(p!=q && q->urm!=p)  
{  
    p=p->urm;  
    q=q->prec;  
    n=n+1;  
}
```

- Care este numărul maxim de elemente pe care le poate avea lista astfel încât, după executarea secvenței alăturate, valoarea variabilei **n** să fie 3? (6p.)

5. Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** ($1 \leq n \leq 6$) și elementele unui tablou bidimensional **A** cu **n** linii și **n** coloane, care memorează numere naturale nenule mai mici decât 100, și afișează pe ecran produsul numerelor "pivot" pentru matricea **A** dacă există astfel de numere, altfel va afișa mesajul **NU EXISTA**. Un număr natural **x** este "pivot" pentru matricea **A** dacă înmulțind fiecare element de pe prima coloană cu numărul **x**, se obțin, în aceeași ordine, elementele unei alte coloane din matrice.

2	7	4	8	4
1	1	2	4	2
3	12	6	12	3
1	22	2	4	2
5	10	10	20	8

Exemplu: pentru matricea din figura alăturată se afișează 8.

(10p.)

Subiectul III (30 de puncte)

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

1. În câte dintre permutările elementelor mulțimii $\{ 'I', 'N', 'F', 'O' \}$ vocalele apar pe poziții consecutive? (4p.)
- a. 24 b. 6 c. 12 d. 4

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

2. Ce se afișează ca urmare a apelului `p(123)`; dacă subprogramul `p` are definiția alăturată? (6p.)
- ```
void p (int x)
{cout<<x; | printf("%d",x);
 if(x!=0){p(x/10);
 cout<<x%10; | printf("%d",x%10);}}
```

3. Scrieți definiția completă a subprogramului `multipli`, cu trei parametri `a, b, c` ( $a \leq b$ ), numere naturale din interval  $[1, 10000]$  ce returnează numărul multiplilor lui `c` din intervalul  $[a; b]$ .

**Exemplu:** pentru  $a=10, b=27, c=5$  subprogramul returnează valoarea 4. (10p.)

4. Se consideră două tablouri unidimensionale **A** și **B** cu elemente numere naturale din intervalul  $[1, 10000]$ . Spunem că tabloul **A** "se poate reduce" la tabloul **B** dacă există o împărțire a tabloului **A** în secvențe disjuncte de elemente aflate pe poziții consecutive în tabloul **A** astfel încât prin înlocuirea secvențelor cu suma elementelor din secvență să se obțină, în ordine, elementele tabloului **B**.

De exemplu tabloul

|   |    |   |   |   |   |    |   |   |    |   |   |   |
|---|----|---|---|---|---|----|---|---|----|---|---|---|
| A | 7  | 3 | 4 | 1 | 6 | 4  | 6 | 9 | 7  | 1 | 8 | 7 |
| B | 14 |   |   | 7 |   | 26 |   |   | 16 |   |   |   |

se poate reduce la tabloul

Fișierul text **NUMERE.IN** conține pe prima linie două numere naturale nenule  $n$  și  $m$  ( $1 \leq m \leq n \leq 100$ ), pe linia a doua  $n$  numere naturale din intervalul  $[1; 10000]$  și pe linia a treia alte  $m$  numere naturale din intervalul  $[1; 10000]$ . Pe fiecare linie numerele sunt separate prin câte un spațiu.

a) Scrieți un program **C/C++** care citește toate numerele din fișierul **NUMERE.IN** și verifică, utilizând un algoritm eficient din punctul de vedere al timpului de executare, dacă tabloul construit cu cele  $n$  numere aflate pe linia a doua în fișier se poate reduce la tabloul construit cu cele  $m$  numere aflate pe linia a treia în fișier. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz negativ. (6p.)

b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)