

Subiectul II (30 de puncte)

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

1. Se consideră graful neorientat cu mulțimea nodurilor {1, 2, 3, 4, 5, 6, 7, 8} și mulțimea muchiilor {[1, 2], [2, 3], [2, 4], [4, 7], [2, 6], [1, 5], [5, 6], [6, 8], [7, 8]}. Pentru a transforma graful într-un arbore, putem elimina: (4p.)
- a. muchiile [1, 5] și [1, 2] b. muchia [5, 6]
c. nodul 3 d. muchiile [2, 6] și [4, 7]
2. Se consideră o listă alocată dinamic care are cel puțin 10 elemente și fiecare element al listei memorează în câmpul info un număr întreg, iar în câmpul next adresa elementului următor în listă sau NULL dacă nu există un element următor. Variabila p memorează adresa de început a listei, iar variabila aux este de același tip cu p. Dacă în urma executării secvenței alăturate de program, variabila p are valoarea NULL, atunci: (4p.)
- ```
while (p!=NULL && p->info%5!=0)
{ aux=p;
 p=aux->next;
 delete p; | free(p);
}
```
- a. toate numerele din listă sunt divizibile cu 5                      b. doar primul element din listă este divizibil cu 5  
c. în listă nu s-a memorat niciun număr divizibil cu 5                      d. doar ultimul element memorat în listă este divizibil cu 5

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

3. Ce se va afișa în urma executării secvenței alăturate, în care variabila c memorează un șir cu cel mult 20 de caractere, iar variabila i este de tip întreg? (6p.)
- ```
char c[]="tamara";
for(i=0;i<3;i++)
  c[i]=c[i+1];
cout<<c; | printf("%s",c)
```
4. Un graf neorientat cu 10 noduri, numerotate de la 1 la 10, este reprezentat cu ajutorul listelor de adiacență alăturate. Câte componente conexe are graful și care este numărul minim de muchii ce trebuie adăugate pentru ca graful să fie conex? (6p.)
- | | |
|---------|-------|
| 1: 3, 5 | 6: |
| 2: 4 | 7: 10 |
| 3: 1, 5 | 8: 4 |
| 4: 2, 8 | 9: |
| 5: 1, 3 | 10: 7 |
5. Scrieți programul C/C++ care citește de la tastatură un număr natural n (n ≤ 50) și construiește în memorie o matrice cu n linii și n coloane, ale cărei elemente sunt numere întregi citite de la tastatură. Pentru fiecare coloană a matricei, în ordine, programul afișează pe ecran cel mai mic număr de pe respectiva coloană. Numerele afișate vor fi separate prin câte un spațiu. (10p.)
- Exemplu: pentru n=4 și matricea alăturată, se vor afișa pe ecran valorile: -7 18 -10 2.
- | | | | |
|-----|-----|-----|----|
| 122 | 103 | 5 | 10 |
| -7 | 18 | -10 | 2 |
| 107 | 999 | 59 | 4 |
| 1 | 200 | 100 | 7 |

Subiectul III (30 de puncte)

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

1. Pentru generarea tuturor mulțimilor de câte 5 cifre, având la dispoziție cifrele de la 1 la 9, se poate utiliza un algoritm echivalent cu algoritmul de generare a: (4p.)
- a. permutărilor de 5 elemente b. submulțimilor mulțimii {1, 2, 3, 4, 5, 6, 7, 8, 9}
- c. combinațiilor de 9 elemente luate câte 5 d. aranjamentelor de 9 elemente luate câte 5

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

2. Subprogramul **f** este definit alăturat. Ce se va afișa în urma executării secvenței de mai jos, în care variabilele **a** și **b** sunt de tip întreg? (6p.)
- ```
a=3; b=9; f(b,a);f(b,b);
```

```
void f(int &a,int b)
{
 a=a-5;b=a-2;
 cout<<a<<b; | printf("%d%d",a,b);
}
```
3. Fișierul text **numere.in** conține pe prima linie un număr natural **n** ( $0 < n < 1000$ ), iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program **C/C++** care citește toate numerele din fișier și afișează pe ecran, separate prin câte un spațiu, numerele de pe a doua linie a fișierului, care încep și se termină cu aceeași cifră. (10p.)

**Exemplu:** dacă fișierul **numere.in** are conținutul alăturat, se numerele ce se vor afișa sunt: **55 3 101 7 2782**

```
9
55 107 3 101 92 7 208 2782
80
```

4. a) Scrieți definiția completă a unui subprogram **sum** care primește prin parametrul **x** un număr natural de cel mult 4 cifre și returnează suma divizorilor numărului **x**, diferiți de 1 și de el însuși.

**Exemplu:** dacă **x=10** se va returna valoarea **7** ( $7=2+5$ ). (4p.)

b) Scrieți programul **C/C++** care citește de la tastatură un număr natural **n** ( $0 < n < 100$ ), apoi **n** numere naturale (cu cel mult 4 cifre fiecare). Programul determină, folosind apeluri utile ale subprogramului **sum**, pentru fiecare număr natural citit, suma divizorilor săi proprii și afișează pe ecran sumele determinate, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu. (6p.)

**Exemplu:** dacă **n=5** și numerele citite sunt **10 2 33 6 11**

valorile afișate pe ecran vor fi: **0 0 5 7 14**

deoarece suma divizorilor lui **10** este **7**, suma divizorilor lui **2** este **0**, suma divizorilor lui **33** este **14**, suma divizorilor lui **6** este **5**, suma divizorilor lui **11** este **0**.