

**Examenul național de bacalaureat**  
**Proba E. d)**  
**INFORMATICĂ**  
**Limbajul C/C++**

*Filieră teoretică, profil real, specializare matematică-informatică/matematică-informatică, intensiv informatică.*  
*Filieră vocațională, profil militar, specializare matematică-informatică.*

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei ore.
- **Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.**
- **În grafurile din cerințe oricare muchie are extremități distincte și oricare două muchii diferă prin cel puțin una dintre extremități.**

**SUBIECTUL I****(20 de puncte)**

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Care dintre următoarele instrucțiuni afișează doar numerele pare de la 1 la 10?

- a. `for(int i=1; i<=10; i++) cout<<i<<" "`;
- b. `for(int i=2; i<=10; i+=2) cout<<i<<" "`;
- c. `for(int i=1; i<=10; i+=2) cout<<i<<" "`;
- d. `for(int i=0; i<10; i++) cout<<i<<" "`;

2. Se consideră un graf orientat cu 5 vârfuri numerotate de la 1 la 5 și arcele:

(1,2), (2,3), (3,1), (3,4), (4,5), (5,4)

Indicați numărul vârfurilor pentru care gradul interior este egal cu gradul exterior.

- a. 1
- b. 2
- c. 3
- d. 4

3. Ce valoare se afișează în urma rulării următoarei secvențe de program?

```
int s = 1;
for(int i = 1; i <= 3; i++)
    s = s * i;
cout << s;
```

- a. 3
- b. 6
- c. 9
- d. 12

4. Se generează toate numerele naturale de 3 cifre, formate din cifre de la 1 la 5, astfel încât cifrele să fie în ordine strict crescătoare. Câte astfel de numere există?

- a. 5
- b. 10
- c. 15
- d. 20

5. Care este valoarea afișată de următorul program?

```
int v[7] = {3, 1, 4, 1, 5, 9, 2};
```

```
int s = 0;
```

```
for(int i = 0; i < 7; i++) {
```

```
    if(v[i] % 2 == 1)
```

```
        s = s + v[i];
```

```
    else
```

```
        s = s - v[i] / 2;
```

```
}
```

```
cout << s;
```

- a. 16
- b. 12
- c. 20
- d. 14

## SUBIECTUL II

(40 de puncte)

1. Se consideră algoritmul alăturat descris în pseudocod.

Se știe că printre elementele vectorului se află cel puțin un număr par și unul impar.

a) Scrieți valoarea care se va afișa dacă se citesc, în această ordine, numerele 6,2,20,22,2,5,11?

(6p.)

```
citeste n(număr natural
nenul)
nr←0
s←0
pentru i←0, n-1 execută
    citeste v[i](numere
întregi)
    dacă (v[i]%2!=1)
atunci
    nr←nr+1
    s←s+v[i]
scrie s/nr doar partea
întreagă)
```

b) Pentru  $n=5$ , determinați un set de cinci numere din intervalul  $[10,100]$  care pot fi citite, astfel încât, în urma executării algoritmului, să se afișeze valoarea 12.

(6p.)

c) Să se scrie programul C/C++ corespunzător.

(6p.)

d) Să se scrie un algoritm echivalent cu algoritmul dat care să înlocuiască structura repetitivă cu număr cunoscut de pași cu o structură repetitivă cu test final.

(6p.)

2. Dacă  $i$  este o variabilă de tip întreg, iar  $s$  reține un șir cu cel mult 31 caractere, ce se va afișa în urma executării secvenței alăturate?

```
strcpy(s,"carte");
for(i=0;i<strlen(s);i++)
    if(s[i]=='c')
        s[i]=s[3];
cout<<s;
```

(6p.)

3. Scrieți un program care citește de la tastatură un număr natural  $n(3 \leq n \leq 50)$  și elementele unui tablou bidimensional cu  $n$  linii și  $n$  coloane, numere naturale cu cel mult 4 cifre, apoi modificați tabloul în memorie, eliminând elementele aflate pe diagonala principală, ca în exemplu. Programul trebuie să afișeze pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele de pe aceeași linie fiind separate prin câte un spațiu.

Exemplu: pentru  $n=4$  și tabloul:

7 5 3 4	se va afișa	5 3 4
8 1 5 6		8 5 6
4 6 3 5		4 6 5
0 9 9 6		0 9 9

(10p.)

**SUBIECTUL III**

( 30 de puncte)

1. Subprogramul **numărare** are doi parametri:

- $n$ , prin care primește un număr natural ( $1 \leq n \leq 10^9$ );
- $m$ , prin care primește un număr natural ( $1 \leq m \leq 10^9$ ,  $n \leq m$ );

Subprogramul returnează numărul de cifre utilizate pentru construirea numerelor naturale din intervalul  $[n, m]$ . Scrieți definiția completă a subprogramului **numărare**.

**Exemplu:** Dacă  $n=7$  și  $m=56$  atunci, în urma apelului, subprogramul va returna valoarea **115**. (10 p.)

2. Scrieți un program C/C++ care citește de la tastatură două numere naturale  $m$  și  $n$  ( $2 \leq m \leq 50$ ,  $2 \leq n \leq 50$ ) și elementele unui tablou bidimensional cu  $m$  linii și  $n$  coloane, numere naturale cu cel mult patru cifre fiecare. Programul determină coloanele care au toate elementele egale cu aceeași valoare și, pentru fiecare astfel de coloană afișează pe ecran valoarea respectivă. Valorile afișate sunt separate

prin câte un spațiu, iar dacă nu există astfel de valori, programul afișează pe ecran mesajul **NU EXISTA**. (10 p.)

**Exemplu:** pentru  $m=4$ ,  $n=5$  și tabloul alăturat, se afișează pe ecran numerele **3 7 7**, nu neapărat în această ordine.

2	3	7	5	7
3	3	7	4	7
7	3	7	1	7
8	3	7	6	7

3. Fișierul **bac.in** conține pe prima linie un șir de cel mult **100 000** de numere întregi formate din cel mult **2** cifre fiecare. Numerele din șir sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran, separate printr-un spațiu, numărul sau numerele din fișier cu număr maxim de apariții.

**Exemplu:** dacă fișierul **bac.in** conține numerele **27 -8 43 27 -8 9 10 43**, atunci pe ecran se vor afișa, nu neapărat această ordine, numerele: **-8 43 27**.

a. Scrieți programul **C/C++** corespunzător algoritmului proiectat. (8 p.)

b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia din punct de vedere al timpului de executare și al memoriei utilizate. (2 p.)