

**EXAMENUL DE BACALAUREAT – 2008**  
**Proba scrisă la INFORMATICĂ**  
**PROBA E, limbajul C/C++**  
**Specializarea Matematică-informatică intensiv informatică**

- ♦ Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- ♦ Timpul efectiv de lucru este de 3 ore.
- ♦ În rezolvările cerute, identificatorii utilizați 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).

**Subiectul I (30 de puncte)**

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

1. În secvența de instrucțiuni **C/C++** alăturată, variabilele **x** și **y** sunt întregi. Ce valoare va reține variabila **x** după executarea acesteia? (4p.)
- |        |       |       |       |
|--------|-------|-------|-------|
| a. -10 | b. 25 | c. 15 | d. 40 |
|--------|-------|-------|-------|

```
x=20; y=5;  
x=x+y;  
y=x-2*y;  
x=y+x;
```

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

**2. Se consideră algoritmul alăturat, descris în pseudocod.**

S-a notat cu **x%y** restul împărțirii numărului natural **x** la numărul natural nenul **y** și cu **[z]** partea întreagă a numărului real **z**.

- a) Scrieți valoarea afișată dacă pentru **x** se citește valoarea **4589** și pentru **y** se citește valoarea **723**. (6p.)
- b) Scrieți ce valori pot fi citite pentru **x** și **y** astfel încât valoarea afișată să fie **200**. (4p.)
- c) Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat în care să se înlocuiască structura **repetă...până când** cu o structură repetitivă cu test inițial. (6p.)

```
citește x,y  
      (numere naturale nenule)  
t←0  
u←1  
repetă  
| dacă x%10 > y%10  
|   atunci  
|   | z ← x%10  
|   altfel  
|   | z ← y%10  
|   ■  
t←t+z*u  
u←u*10  
x←[x/10]  
y←[y/10]  
până când x=0 și y=0  
scrie t
```

**Subiectul II (30 de puncte)**

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

- 1.** Cum se poate accesa prima literă a numelui unui elev a cărui date de identificare sunt memorate în variabila `e`, declarată alăturat? (4p.)

```
struct elev{  
    char nume[20], prenume[20];  
    int varsta;}e;
```

  - a. `e->nume[0]`
  - b. `e.nume[0]`
  - c. `elev.nume[0]`
  - d. `nume.e[0]`  

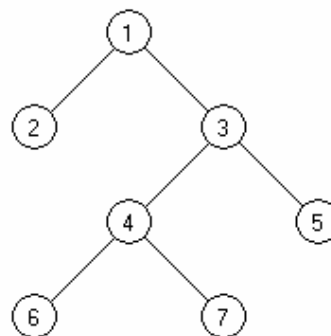
**2.** Se consideră graful orientat dat prin matricea de adiacență alăturată. Care este numărul de vârfuri ale grafului care au gradul interior (intern) egal cu gradul exterior (extern)? (4p.)

	0	0	0	0	0
	1	0	1	1	1
	0	0	0	1	0
	1	0	0	0	1
	0	1	0	0	0

  - a. 0
  - b. 3
  - c. 2
  - d. 1

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

3. Care este vectorul de "tați" asociat arborelui cu rădăcină din figura alăturată? (6p.)



4. Se consideră o listă liniară simplu înlanțuită ale cărei noduri rețin în câmpul **next** adresa nodului următor sau **NULL** dacă nu există un element următor în listă. Lista are cel puțin un element. Știind că variabila **u** reține adresa ultimului nod din listă, care este secvența de instrucțiuni în limbajul **C/C++** prin care se inserează în listă după nodul **u** un nou nod a cărui adresă este reținută de variabila **v**, de același tip cu **u**? (6p.)
5. Scrieți programul **C/C++** care citește de la tastatură un număr natural **n** ( $1 \leq n \leq 50$ ) și **n \* n** numere naturale de cel mult 5 cifre ce reprezintă elementele unui tablou bidimensional **a**, cu **n** linii și **n** coloane, și verifică dacă matricea este triunghiulară superior. Programul va afișa pe ecran mesajul corespunzător: „**Este triunghiulară superior**” respectiv „**Nu este triunghiulară superior**”. O matrice se numește triunghiulară superior dacă toate elementele aflate sub diagonala principală a ei sunt nule. (10p.)

**Exemplu:** pentru  $n=3$  și matricea alăturată se va afișa mesajul:

Este triunghiulară superior

1	2	3
0	5	6
0	0	9

**Subiectul III (30 de puncte)**

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

1. Construiam anagramele unui cuvânt  $c_1c_2c_3c_4$  prin generarea în ordine lexicografică a permutărilor indicilor literelor cuvântului și obținem  $c_1c_2c_3c_4$   $c_1c_2c_4c_3$   $c_1c_3c_2c_4$  ...  $c_4c_3c_1c_2$   $c_4c_3c_2c_1$ . Pentru anagramele cuvântului **rateu**, după șirul **raetu**, **raeut**, **raute** cuvintele imediat următoare sunt:
- (4p.)
- |                                  |                                 |
|----------------------------------|---------------------------------|
| a. <b>rau</b> et și <b>rtaeu</b> | b. <b>rtaeu</b> și <b>rtaue</b> |
| c. <b>rau</b> et și <b>rtaue</b> | d. <b>rtaeu</b> și <b>ratue</b> |

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

2. Se consideră subprogramul recursiv definit alăturat. Ce valori vor fi afișate pe ecran în urma apelului **beta(15)**? (6p.)

```
void beta(int n)
{if (n!=1)
    {printf("%d ",n); | cout<<n<<" ";
    if (n%3==0) beta(n/3);
    else beta(2*n-1);}
else printf("%d",1); | cout<<1;}
```

3. Să se scrie în limbajul **C/C++** definiția completă a subprogramului **calcul**, care primește prin intermediul parametrului întreg **n**, un număr natural de cel mult 9 cifre și returnează valoarea absolută a diferenței dintre numărul de cifre pare și numărul de cifre impare conținute de **n**. (10p.)

**Exemplu :** în urma apelului **calcul(92465)** se va returna valoarea 1.

4. Fișierul text **NUMERE.TXT** conține pe prima linie un număr natural **n** ( $1 \leq n \leq 10000$ ) și pe a doua linie, separate prin spații, **n** numere naturale cu cel mult 9 cifre fiecare. Aceste numere sunt dispuse în ordine **crescătoare** și separate între ele printr-un spațiu.

a) Scrieți un program **C/C++** care citește valorile din fișier și, printr-o metodă eficientă din punct de vedere al timpului de executare și al spațiului de memorie utilizat, afișează pe ecran separate printr-un spațiu, în ordine crescătoare, numerele pare de pe a doua linie a fișierului, urmate de cele impare în ordine descrescătoare. (6p.)

**Exemplu:** dacă fișierul **NUMERE.TXT** are

6	212	412	5111	71113	81112	101112
---	-----	-----	------	-------	-------	--------

conținutul alăturat

atunci programul va afișa pe ecran **212 412 81112 101112 71113 5111**

b) Descrieți succint, în limbaj natural, metoda utilizată la punctul a, justificând eficiența acesteia. (4p.)