

**EXAMENUL DE BACALAUREAT – 2009**  
**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. Variabile întregi **x** și **y** memorează câte un număr natural, cu exact două cifre. Care este valoarea expresiei **x-y** știind că fiecare dintre expresiile C/C++ alăturate are valoarea 1? (4p.)
- |      |      |      |       |
|------|------|------|-------|
| a. 0 | b. 9 | c. 1 | d. 11 |
|------|------|------|-------|
- x/10==y%10**

**y/10==x%10**

**x/10==x%10+1**

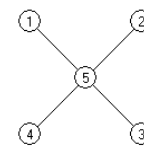
**Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.**

2. Se consideră algoritmul alăturat descris în pseudocod. (6p.)
- a) Scrieți valoarea care se afișează dacă se citesc numerele **n=2** și **m=11**. (6p.)
- b) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- c) Dacă pentru **n** se citește valoarea 1 scrieți numărul de valori naturale nenule de exact o cifră, care pot fi citite pentru variabila **m**, astfel încât să se afișeze valoarea 0. (6p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat, care să **NU** folosească structuri repetitive sau recursive. (4p.)
- ```
citește n,m
(numere naturale, n≤m)
s←0
cât timp n<m execută
| s←s+n
| n←n+3
■
dacă n=m atunci
| scrie s+n
altfel
| scrie 0
■
```

**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 minim de noduri ce trebuie eliminate din graful alăturat astfel încât subgraful obținut să **nu** fie conex? (4p.)

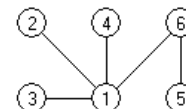


- a. 3                      b. 0                      c. 2                      d. 1
2. În declararea alăturată, câmpurile **x** și **y** ale înregistrării pot memora coordonatele carteziene ale unui punct din planul **xOy**. Care dintre următoarele expresii are valoarea 1 dacă și numai dacă punctul **P** este situat pe axa **ox**? (6p.)
- a. **P.x==0**                      b. **P.y==0**                      c. **P.x+P.y==0**                      d. **P.x==P.y**

```
struct punct
{
    float x,y;
}P;
```

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

3. Se consideră arborele din figura alăturată.
- a) Care este nodul ce trebuie ales ca rădăcină astfel încât aceasta să aibă 4 descendenți direcți (fii)? (3p.)
- b) Care sunt cei patru fii ai nodului ales ca rădăcină în acest caz? (3p.)



4. O listă liniară simplu înlanțuită cu 99 de elemente, reține în câmpul **nr** al fiecărui element câte un număr natural, iar în câmpul **urm**, adresa elementului următor din listă sau **NULL** dacă nu există un element următor. Știind că **prim** păstrează adresa primului element al listei și că **p** și **q** sunt două variabile de același tip cu **prim**, câte elemente are lista după executarea secvenței alăturate? (4p.)
- ```
p=prim;
while(p->urm!=NULL)
{q=p->urm;
 p->urm=q->urm;
 delete q; | free(q);
 p=p->urm;}
```
5. Un șir de caractere **s** se numește “șablon” pentru un alt șir de caractere **x**, dacă este format din caractere din mulțimea **{\*, ?, #}**, are aceeași lungime cu **x** și pe fiecare poziție din **s** în care apare **\*** în **x** se găsește o vocală, pe fiecare poziție din **s** în care apare **#** în **x** se găsește o consoană și pe fiecare poziție din **s** în care apare **?** putem avea orice caracter în **x**. Se consideră vocală orice literă din mulțimea **{a, e, i, o, u}**. Scrieți programul **C/C++** care citește de la tastatură două șiruri de caractere, de aceeași lungime, formate din cel mult 200 de litere mici ale alfabetului englez, și afișează pe ecran, un șablon **comun** celor două șiruri citite, care conține un număr minim de caractere **?**. Exemplu: pentru șirurile **diamant** și **pierdut** se afișează **\*\*\*#??#** (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  $\{ 'I', 'N', 'F', 'O' \}$  vocala 'I' apare pe prima poziție? (4p.)
- a. 1                                      b. 24                                      c. 6                                      d. 12

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

2. Subprogramul **f** este definit alăturat. Ce se afișează ca urmare a apelului **f(1,3)**? (6p.)
- ```
void f (int x,int y)
{   if (x<y){x=x+1;f(x,y);
      y=y-1;f(x,y);
    }
    else
      cout<<x<<y; printf("%d%d",x,y);
}
```
3. Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 99$ ), impar, și construiește în memorie un tablou unidimensional  $A=(A_1, A_2, \dots, A_n)$  cu elementele mulțimii  $\{1, 2, \dots, n\}$  astfel încât elementele de pe poziții impare formează șirul crescător  $1, 2, \dots, [(n+1)/2]$ , iar elementele de pe poziții pare șirul descrescător  $n, n-1, \dots, [(n+1)/2]+1$ .

**Exemplu:** pentru  $n=11$  se va construi tabloul **A** :

|   |    |   |    |   |   |   |   |   |   |   |
|---|----|---|----|---|---|---|---|---|---|---|
| 1 | 11 | 2 | 10 | 3 | 9 | 4 | 8 | 5 | 7 | 6 |
|---|----|---|----|---|---|---|---|---|---|---|

Programul va crea un fișier text **TABLOU.TXT**. Elementele tabloului se vor scrie, în ordine, pe prima linie a fișierului, cu câte un spațiu între ele. (10p.)

4. a) Scrieți definiția completă a subprogramului **dcm**, cu doi parametri, care:
- primește prin parametri **a** și **b** două valori naturale din intervalul  $[1, 30000]$
  - returnează o valoare naturală reprezentând cel mai mare număr care este atât divizor al lui **a** cât și divizor al lui **b**.

**Exemplu:** dacă  $a=100$  și  $b=120$ , subprogramul returnează valoarea 20. (6p.)

b) Scrieți un program C/C++ care citește de la tastatură două numere naturale **a** și **b**, numere din intervalul  $[1, 30000]$  și determină, folosind apeluri utile ale subprogramului **dcm**, cel mai mare număr care este divizor al fiecăruia dintre numerele **a** și **b** și are proprietatea că este un produs de două sau mai multe numere prime distincte. Programul afișează pe ecran numărul cu proprietatea cerută, iar dacă nu există un astfel de număr, afișează mesajul **nu exista**.

**Exemplu:** dacă  $a=60$  și  $b=72$ , atunci se afișează 6

iar dacă  $a=100$  și  $b=75$ , atunci se afișează **nu exista**. (4p.)