

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. Care dintre următoarele expresii C/C++ are valoarea 1 dacă numărul natural memorat de variabila **n** este strict mai mare decât 100 și divizibil cu 12? (4p.)
- a. `(n%4 == 0) || (n%3 == 0)` b. `(n%100/4 == 0) && !(n%3)`
c. `(n%100%4 == 0) && n%3` d. `(n%100%4 == 0) && (n%3 == 0)`

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

2. Se consideră algoritmul alăturat, descris în pseudocod.

- a) Scrieți ce se va afișa dacă pentru **n** se citește valoarea 5. (6p.)
- b) Se înlocuiește prima structură **dacă...atunci** cu atribuirea **j←4**. Modificați condiția logică din cadrul structurii **cât timp...execută** astfel încât, pentru **n=4**, algoritmul să afișeze:

**
* (4p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```
citește n (număr natural nenul)
pentru i←1, 2*n-1 execută
    b ← 0
    dacă n-i < 0 atunci
        j←i-n
    altfel
        j←n-i
    ■
    cât timp j ≥ 0 execută
        scrie "*"
        j←j-1
        b←1
    ■
    dacă b = 0 atunci
        salt la rând nou (sfârșit de rând)
    ■
■
```

- d) Scrieți în pseudocod un algoritm echivalent cu cel dat în care să se înlocuiască prima structură **dacă...atunci** cu o operație de atribuire. (6p.)

Subiectul II (30 de puncte)

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

1. Fie un arborele cu rădăcină cu 9 noduri, numerotate de la 1 la 9. Care este vectorul „de tați” al acestui arbore știind că nodurile 1, 2, 3, 4, 5, 6, 7, 8 au exact câte un descendent direct (fiu)? (4p.)
- a. (1, 2, 3, 4, 5, 6, 7, 8) b. (1, 2, 3, 4, 5, 6, 7, 8, 9)
c. (0, 1, 2, 3, 4, 5, 6, 7, 8) d. (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
2. În secvența alăturată, fiecare dintre variabilele **x** și **s** sunt de tipul șir de caracter, iar **i** este de tip întreg. Dacă variabilele **x** și **s** memorează inițial șirul **absolvent**, ce se va memora în variabila **x** în urma executării secvenței alăturate? (4p.)
- ```
for(i=0; i<strlen(s); i++)
 if(strcmp(x, s+i)<0)
 strcpy(x, s+i);
```
- a. nt                      b. absolvent                      c. solvent                      d. vent

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

3. Se consideră un graf neorientat cu 8 noduri, numerotate de la 1 la 8, și muchiile: [1, 4], [1, 8], [2, 1], [2, 3], [3, 1], [4, 5], [4, 7], [5, 7], [6, 5]. Scrieți câte componente conexe are graful dat și care dintre nodurile acestuia trebuie eliminate astfel încât subgraful obținut să aibă un număr maxim de componente conexe. (6p.)
4. Se consideră lista simplu înlănțuită în care fiecare nod memorează în câmpul **info** un număr întreg, iar în câmpul **urm** adresa nodului următor. În listă sunt memorate, în această ordine, numai valorile 1, 2, 3. Dacă variabila **p** reține adresa primului nod din listă, iar variabila **u** adresa ultimului nod din listă, scrieți instrucțiunile care pot înlocui zona punctată din secvența alăturată, astfel încât, în urma executării acesteia, să se afișeze 3 2 1. (6p.)
- ```
.....  
.....  
while(u != NULL)  
{  
    cout << u->info<<" ";  
    u = u->urm;  
}
```
5. Scrieți programul **C/C++** care citește de la tastatură o valoare naturală **n** ($2 \leq n \leq 100$), construiește în memorie și apoi afișează pe ecran o matrice **a**, cu **n** linii și **n** coloane, **simetrică** față de diagonala secundară. Elemente matricei situate deasupra diagonalei secundare, inclusiv diagonala secundară, sunt toate numerele naturale de la 1 la $\frac{n(n+1)}{2}$.
- Elementele fiecărei linii, de la stânga la dreapta și ale fiecărei coloane, de sus în jos, situate deasupra diagonalei secundare, inclusiv cele de pe diagonala secundară, sunt în ordine strict crescătoare, ca în exemplu. Matricea se va afișa pe ecran, câte o linie a matricei pe o linie a ecranului, elementele fiecărei linii fiind separate între ele prin câte un spațiu.
- Exemplu:** pentru **n = 4** se va afișa matricea alăturată. (10p.)
- | | | | |
|---|---|---|----|
| 1 | 3 | 6 | 10 |
| 2 | 5 | 9 | 6 |
| 4 | 8 | 5 | 3 |
| 7 | 4 | 2 | 1 |

Subiectul III (30 de puncte)

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

1. Se generează în ordine crescătoare toate numerele de 4 cifre, cu cifre distincte, astfel încât diferența în valoare absolută dintre prima și ultima, respectiv a doua și a treia cifră este egală cu 2. Primele 11 soluții generate sunt, în ordine: 1023, 1203, 1243, 1423, 1463, 1573, 1643, 1683, 1753, 1793, 1863. Care dintre următoarele numere se va genera imediat înaintea numărului 9317? (4p.)
- a. 9247 b. 9357 c. 9207 d. 8976

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

2. Scrieți ce se va afișa în urma executării subprogramului alăturat, la apelul **F(57)**. (6p.)
- ```
void F(int x)
{
 if(x != 0)
 {
 F(x/2);
 cout << x%2; | printf("%d", x%2);
 }
}
```
3. Scrieți definiția completă a subprogramului **Ecuatie** care primește prin parametri **a**, **b** și **c** trei numere întregi,  $a \neq 0$ , de cel mult patru cifre fiecare, reprezentând coeficienții ecuației de gradul al II-lea:  $ax^2 + bx + c = 0$ . În funcție de soluțiile ecuației subprogramul va returna:
- cea mai mare dintre soluții dacă ecuația are două soluții reale distincte, dintre care cel puțin una pozitivă.
  - una dintre soluții dacă ecuația are două soluții egale și pozitive.
  - -32000 în celelalte cazuri.
- (10p.)
4. Se consideră șirul **s**: 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 1, 2, ... . Pentru un număr natural **k**,  $0 < k \leq 100000$ , se cere să se determine valoarea elementului ce se află pe poziția **k** în șirul **s**.  
**Exemplu:** pentru **k=5** numărul cerut este 2.
- a) Scrieți un program C/C++ care citește de la tastatură valoarea numărului natural **k** și, prin apeluri utile ale funcției **Ecuatie**, definite la punctul 3, determină valoarea elementului ce se află pe poziția **k** în șirul **s**, folosind un algoritm eficient din punctul de vedere al spațiului de memorie alocat și al timpului de executare. Valoarea astfel determinată se va scrie în fișierul text **sir.out**. (6p.)
- b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri) (4p.)