

**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. Știind că variabila întreagă **nr** memorează valoarea 5, stabiliți ce mesaj se va afișa în urma executării secvenței de instrucțiuni C/C++ următoare. (4p.)

<pre>//C++ if (nr&lt;7) if (nr&gt;3)     cout&lt;&lt;"Bine"; else cout&lt;&lt;"Foarte bine"; else cout&lt;&lt;"Rau";</pre>	<pre>//C if (nr&lt;7) if (nr&gt;3)     printf("Bine"); else printf("Foarte bine"); else printf("Rau");</pre>
--	--

- a. BineRau                      b. Foarte bine                      c. Rau                      d. Bine

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

2. Se consideră algoritmul alăturat descris în pseudocod.

S-a notat cu **[a]** partea întreagă a numărului real **a** și cu **|b|** valoarea absolută a numărului întreg **b**.

- a) Scrieți valoarea care se va afișa pentru **z=50**. (6p.)
- b) 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.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Scrieți de câte ori se execută atribuirea **y←x** în timpul executării algoritmului obținut din cel dat prin înlocuirea inițializării **x←1** cu **x← $\lceil \sqrt{z} \rceil$** . (4p.)

```
citește z
      (număr întreg nenul)
z←|z|
x←1
repetă
    y←x
    x←[(x+z/x)/2]
până când x=y
scrie x
```

**Subiectul II (30 de puncte)**

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

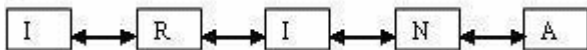
1. Se consideră un graf neorientat  $G$  cu 101 noduri și 101 muchii. Numărul maxim de vârfuri izolate ale grafului poate fi: (4p.)  
a. 0                      b. 10                      c. 50                      d. 86
2. Un arbore cu 11 noduri, numerotate de la 1 la 11, este memorat cu ajutorul vectorului de tați  $t = (2, 5, 5, 3, 0, 2, 4, 6, 6, 2, 3)$ . Descendenții direcți (fiii) ai nodului 2 sunt: (4p.)  
a. 1, 6 și 10              b. 5                      c. 6, 8 și 9              d. 3

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

3. Se consideră definită matricea  $A$  care are 10 linii, numerotate de la 1 la 10, și 8 coloane, numerotate de la 1 la 8, ale cărei elemente sunt numere întregi. Rescrieți secvența alăturată astfel încât toate elementele de pe cea de-a treia coloană a matricei să fie inițializate cu valoarea 100. (6p.)  

```
for (j=1; j<=10; j++)  
    A[j][8]=0;
```
4. Ce se va afișa în urma executării secvenței alăturate de program, în care variabila  $c$  memorează un șir cu cel mult 20 de caractere, iar variabila  $i$  este de tip întreg? (6p.)  

```
char c[]="abracadabra";  
i=6;  
cout<<c[i]<<c[i+1] <<endl;  
| printf("%c%c\n", c[i],c[i+1]);  
while (i>=0)  
    {cout<<c[i]; | printf("%c",c[i]);  
      i=i-1;}
```
5. O listă dublu înlănțuită memorează în fiecare nod al său, în câmpul **info** un caracter, iar în câmpurile **prec** și **urm** adresa nodului precedent, respectiv următor din listă. Scrieți programul **C/C++** care citește de la tastatură un cuvânt (având maximum 20 de litere), creează lista dublu înlănțuită care va conține în ordine, de la stânga la dreapta, caracterele cuvântului citit (câte o literă în fiecare nod al listei) și afișează pe ecran caracterele din listă, în ordinea inversă a memorării lor. **Exemplu:** dacă s-a citit de la tastatură cuvântul **IRINA** se crează lista de mai jos, apoi se va afișa **ANIRI**.



(10p.)

**Subiectul III (30 de puncte)**

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

1. Pentru a planifica în orarul unei școli, la clasa a XII-a, 4 ore de informatică în zile lucrătoare diferite din săptămână, câte o singură oră pe zi, se poate utiliza un algoritm de generare echivalent cu algoritmul de? **(4p.)**
- a. generare a permutărilor de 4 elemente      b. generare a aranjamentelor de 4 elemente luate câte 5
- c. generare a aranjamentelor de 5 elemente luate câte 4      d. generare a combinațiilor de 5 elemente luate câte 4

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

2. Subprogramul recursiv alăturat este definit incomplet. Scrieți expresia care poate înlocui punctele de suspensie astfel încât, în urma apelului, `f(12)` să se afișeze șirul de valori:  
`12 6 3 1 1 3 6 13.` **(6p.)**
- ```
void f(int i)
{ if ( . . . )
  { printf("%d ", i); | cout<<i<<' ';
    f(i/2);
    printf("%d ", i); | cout<<i<<' ';
  }
}
```
3. Subprogramul **par** primește prin singurul său parametru, **n**, un număr natural nenul cu cel mult 4 cifre și returnează valoarea 1 dacă **n** conține cel puțin o cifră pară, sau returnează valoarea 0 în caz contrar.  
**Exemplu:** pentru **n=723** subprogramul va returna valoarea 1.
- a) Scrieți numai antetul subprogramului **par**. **(2p.)**
- b) Scrieți un program **C/C++** care citește de la tastatură un număr natural nenul **n** cu cel mult 9 cifre, apoi un șir de **n** numere naturale, cu exact 8 cifre fiecare, și afișează pe ecran, numărul de valori din șirul citit care au cel puțin o cifră pară printre primele 4 poziții, considerate de la stânga spre dreapta. Se vor utiliza apeluri utile ale subprogramului **par**.  
**Exemplu:** dacă **n=3**, iar șirul citit este **12386406**, **37152073**, **73543157** se va afișa **2** (numerele **12386406** și **73543157** respectă condiția cerută). **(8p.)**
4. Fișierul **numere.in** conține cel mult 5000 de numere reale, câte unul pe fiecare linie. Se cere să se scrie un program care să citească toate numerele din fișier și să afișeze pe ecran numărul de ordine al primei, respectiv al ultimei linii pe care se află cel mai mare număr din fișier. Cele două numere vor fi separate printr-un spațiu. Alegeți o metodă de rezolvare eficientă din punct de vedere al spațiului de memorare și al timpului de executare.  
**Exemplu:** dacă fișierul are conținutul alăturat, pe ecran se vor afișa numerele **2 6**.
- a) Descrieți succint, în limbaj natural, metoda de rezolvare aleasă, explicând în ce constă eficiența ei. **(4p.)**
- b) Scrieți programul **C/C++** corespunzător metodei descrise. **(6p.)**

```
3.5
7
-4
7
2
7
6.3
5
```