

**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 expresia `c/c++` alăturată variabila ***x*** este de tip întreg. Această expresie: `x%2+(x+1)%2`  
(4p.)
- a. are valoarea 1 pentru orice număr natural ***x***.
  - b. are valoarea 1 dacă și numai dacă ***x*** este un număr par.
  - c. are valoarea 1 dacă și numai dacă ***x*** este un număr impar.
  - d. are o valoare strict mai mare decât 1 pentru orice număr natural ***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 **[*x*]** partea întreagă a numărului real ***x***.

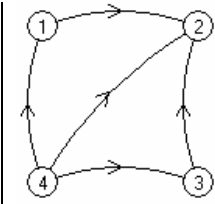
- a) Scrieți valoarea care se afișează dacă se citesc numerele ***a=493*** și ***b=1836***. (6p.)
- b) Scrieți programul `C/C++` corespunzător algoritmului dat. (10p.)
- c) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă de un alt tip. (6p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat, care să **NU** folosească structuri repetitive sau recursive. (4p.)

```
citeste a,b  
                                (numere naturale)  
  
c←0  
d←0  
p←1  
cât timp a+b+c>0 execută  
|   c←a%10+b%10+c  
|   d←d+(c%10)*p  
|   p←p*10  
|   a←[a/10]  
|   b←[b/10]  
|   c←[c/10]  
└─┘  
scrie d
```

**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 arce ce trebuie adăugate în graful orientat din figura alăturată astfel încât fiecare vârf să aparțină unui circuit ? (4p.)



- a. 1                                      b. 2                                      c. 3                                      d. 4
2. Care este numărul nodurilor de tip frunză din arborele cu rădăcină, cu 8 noduri, numerotate de la 1 la 8, reprezentat prin vectorul "de tați" (2, 0, 6, 2, 4, 4, 5, 5) ? (6p.)
- a. 3                                      b. 4                                      c. 5                                      d. 2

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

3. În declarația alăturată, câmpurile **x** și **y** ale înregistrării reprezintă numărătorul, respectiv numitorul unei fracții de forma  $\frac{x}{y}$ .  
Scrieți instrucțiunile prin executarea cărora se memorează în variabila **H** fracția obținută prin adunarea fracțiilor reținute în **F** și **G**. (6p.)

```
struct fractie
{
    int x,y;
} F,G,H;
```

4. O listă liniară simplu înlănțuită, alocată dinamic, 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 astfel de element. Lista memorează, în ordine, numai elementele 1, 2, 3, 4, 5.  
Știind că variabila **prim** memorează adresa primului element al listei și că **p** este o variabilă de același tip cu **prim**, care va fi valoarea reținută în ultimul element după executarea secvenței de mai sus? (4p.)

```
p=prim;
while (p->urm!=NULL)
{p->urm->nr=p->nr*p->urm->nr;
 p=p->urm;
}
```

5. Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** ( $2 < n \leq 10$ ) și construiește în memorie o matrice **A** cu **n** linii și **n** coloane în care orice element aflat pe prima linie, pe prima și pe ultima coloană are valoarea 1 și oricare alt element  $A_{i,j}$  din matrice este egal cu suma a 3 elemente situate pe linia **i-1**: primul aflat pe coloana **j-1**, al doilea pe coloana **j**, iar al treilea pe coloana **j+1**, ca în exemplu. Matricea va fi afișată pe ecran, linie cu linie, numerele de pe aceeași linie fiind separate prin câte un spațiu.  
**Exemplu:** pentru **n=5**, se afișează matricea alăturată. (10p.)

```
1  1  1  1  1
1  3  3  3  1
1  7  9  7  1
1 17 23 17  1
1 41 57 41  1
```

**Subiectul III (30 de puncte)**

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

1. Pentru generarea numerelor cu  $n$  cifre formate cu elementele mulțimii  $\{0, 2, 8\}$  se utilizează un algoritm backtracking care, pentru  $n=2$ , generează, în ordine, numerele 20, 22, 28, 80, 82, 88.  
Dacă  $n=4$  și se utilizează același algoritm, precizați câte numere generate sunt divizibile cu 100? (4p.)
- a. 601                      b. 100                      c. 6                      d. 10

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

2. Subprogramul **scrie** este definit alăturat. (6p.)
- |   |  |
|---|--|
| Ce se afișează ca urmare a apelului <b>scrie(1, 7);</b> | <pre>void scrie (int x, int y) {cout&lt;&lt;x&lt;&lt;y;   printf("%d%d", x, y);   if (x&lt;y)   {scrie(x+1, y-1);    cout&lt;&lt;(x+y)/2;   printf("%d", (x+y)/2);   } }</pre> |
|---|--|
3. Scrieți definiția completă a subprogramului **nreal** cu doi parametri  $x$  și  $y$ , numere naturale din intervalul  $[1; 1000]$  și returnează un număr real cu proprietatea că partea sa întreagă este egală cu  $x$  iar cifrele numărului  $y$ , sunt egale, în ordine, cu cifrele aflate după punctul zecimal.  
**Exemplu:** pentru  $x=12$  și  $y=543$ , subprogramul returnează valoarea 12.543. (10p.)
4. Fișierul **NUMERE.IN** conține pe prima linie un număr natural nenul  $n$  ( $2 \leq n \leq 100$ ) și pe următoarea linie  $n$  numere reale pozitive în ordine strict crescătoare separate prin câte un spațiu.
- a) Scrieți un program **C/C++** care, utilizând un algoritm eficient din punct de vedere al memoriei utilizate, determină și afișează pe ecran cel mai mare număr natural  $x$  cu proprietatea că în orice interval deschis având capete oricare două dintre cele  $n$  numere aflate pe linia a doua în fișierul **NUMERE.IN** se găsesc cel puțin  $x$  numere întregi. Numărul astfel determinat se afișează pe ecran.  
**Exemplu:** dacă fișierul **NUMERE.IN** are conținutul:
- 6
- 3.5 5.1 9.2 16 20.33 100 atunci se afișează 2 pentru că în oricare dintre intervalele  $(3.5; 5.1)$ ,  $(3.5; 9.2)$ ,  $(3.5; 16)$ ,  $(3.5; 20.33)$ ,  $(3.5; 100)$ ,  $(5.1; 9.2)$ ,  $(5.1; 16)$ ,  $(5.1; 20.33)$ ,  $(5.1; 100)$ ,  $(9.2; 16)$ ,  $(9.2; 20.33)$ ,  $(9.2; 100)$ ,  $(16; 20.33)$ ,  $(16; 100)$ ,  $(20.33; 100)$  există cel puțin două numere întregi. (6p.)
- b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)