

**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. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă numărul real memorat în variabila **x** se află în intervalul  $(-2, 2)$ ? **(4p.)**
- a.  **$x*x-4 \leq 0$**       b.  **$4-x*x > 0$**       c.  **$(2 < x) \&\&(x < -2)$**       d.  **$(x-2)*(x+2) > 0$**

**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 întreg **x** la numărul întreg nenul **y**.

- a) Scrieți ce se va afișa dacă se citesc, în această ordine, numerele: 2 5 16 9 12 13 5 0. **(6p.)**
- b) Scrieți un șir de date de intrare, format doar din numere naturale cu o singură cifră fiecare, care să determine afișarea valorii 7310. **(4p.)**

```
citește z,x
(numere naturale nenule)
cât timp x>0 execută
    citește y (număr natural)
    dacă z<y-x atunci
        scrie x%10
    altfel
        scrie y%10
    x←y
```

- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă de alt tip. **(6p.)**
- d) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

**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 orientat cu 6 noduri numerotate de la 1 la 6 și cu mulțimea arcelor formată **doar** din arcele:
- de la fiecare nod numerotat cu un număr neprim  $i$  ( $i > 1$ ) la toate nodurile numerotate cu numere ce aparțin mulțimii divizorilor proprii ai lui  $i$  (divizori diferiți de 1 și de  $i$ )
  - de la nodul numerotat cu 1 la nodul numerotat cu 6
  - de la fiecare nod numerotat cu un număr prim  $i$  la nodul numerotat cu  $i-1$
- Pentru graful dat, care este lungimea celui mai mare drum, format **doar** din noduri distincte? (4p.)
- a. 6                                      b. 5                                      c. 3                                      d. 4
2. Câte frunze are arborele cu rădăcină descris prin următorul vector "de tați":  
(6,5,5,2,0,3,3,3,8,7,7)? (4p.)
- a. 1                                      b. 2                                      c. 5                                      d. 4

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

3. În declararea alăturată, câmpurile  $x$  și  $y$  ale înregistrării pot memora numărătorul, respectiv numitorul unei fracții. Scrieți secvența de instrucțiuni prin executarea căreia se construiește în variabila  $f$  o fracție obținută prin însumarea fracțiilor memorate în variabilele  $f1$  și  $f2$ . (6p.)
- ```
struct fractie
{
    int x,y;
}f,f1,f2;
```

```
struct fractie
{
    int x,y;
}f,f1,f2;
```
4. În secvența de instrucțiuni de mai jos, variabila  $s$  memorează un șir de caractere format doar din litere ale alfabetului englez, iar variabilele  $i$  și  $n$  sunt de tip `int`. Știind că în urma executării secvenței s-a afișat succesiunea de caractere `eied*eael*` scrieți care este șirul de caractere memorat de variabila  $s$ . (6p.)
- ```
//C
n=strlen(s);
for(i=0;i<n;i++)
    if (s[i]=='e') printf("%c",'*');
    else printf("%c%c",'e',s[i]);
```

```
//C++
n=strlen(s);
for(i=0;i<n;i++)
    if (s[i]=='e') cout<<'*';
    else cout<<'e'<<s[i];
```
5. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $2 \leq n \leq 24$ ) și construiește în memorie o matrice cu  $n$  linii și  $n$  coloane ale cărei elemente vor primi valori după cum urmează:
- elementele aflate pe diagonala principală a matricei vor primi valoarea 0
  - elementele de pe prima coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea  $n$
  - elementele de pe a doua coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea  $n-1$
  - ...
  - elementele de pe ultima coloană, cu excepția celui aflat pe diagonala principală vor primi valoarea 1
- Programul va afișa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spațiu între elementele fiecărei linii (ca în exemplu).
- Exemplu:** pentru  $n=4$  se va afișa matricea alăturată. (10p.)
- ```
0 3 2 1
4 0 2 1
4 3 0 1
4 3 2 0
```

```
0 3 2 1
4 0 2 1
4 3 0 1
4 3 2 0
```

**Subiectul III (30 de puncte)**

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

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A = \{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab**, **abac**, **abad**, **abba**, **abbb**, **abbc**, **abbd**, **abbe**. Care este penultimul cuvânt generat? **(4p.)**
- a. edec                      b. eded                      c. edde                      d. edcb

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

2. Pentru definiția alăturată a subprogramului **f**, ce se afișează ca urmare a apelului **f(26);**? **(6p.)**
- ```
void f (int x)
{
    if(x>0)
    if(x%4==0)
    { cout<<'x'; | printf("%c",'x');
      f(x-1); }
    else
    { f(x/3);
      cout<<'y'; | printf("%c",'y');
    }
}
```
3. Fișierului text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult 100 de numere naturale, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește toate numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele din fișier care au cel puțin 3 cifre. Dacă fișierul nu conține astfel de numere se va afișa pe ecran mesajul **NU EXISTA**. **(10p.)**
4. Subprogramul **cif**, cu doi parametri, primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre și prin intermediul parametrului **b** o cifră; subprogramul returnează numărul de apariții ale cifrei **b** în scrierea numărului **a**.  
**Exemplu:** pentru **a=125854** și **b=5**, subprogramul va returna valoarea 2.
- a) Scrieți definiția completă a subprogramului **cif**. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural **n** cu **exact** 8 cifre și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului **cif**, cel mai mare număr palindrom ce poate fi obținut prin rearanjarea tuturor cifrelor numărului **n**. Dacă nu se poate obține un palindrom din toate cifrele numărului **n**, programul va afișa pe ecran numărul 0. Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă.  
**Exemplu:** dacă **n=21523531** atunci se va afișa pe ecran numărul 53211235, iar dacă **n=12272351** atunci se va afișa pe ecran numărul 0. **(6p.)**