

**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. Variabilele **x** și **y** sunt tip **int**. Care dintre expresiile C/C++ de mai jos are valoarea 1 dacă și numai dacă valorile întregi nenule memorate în variabilele **x** și **y** sunt egale? **(4p.)**
- a. **(x%y==0) && (y%x==0) && (x\*y>0)**                      b. **(x==y) && (y<x)**  
c. **(x==y) || (y==x)**    d. **x\*x==y\*y**

**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: 17 22 13 101 2 7 5 0. **(6p.)**

- b) Scrieți un șir de date de intrare, format doar din numere naturale cu cel mult două cifre fiecare, care să determine afișarea valorii 9877. **(4p.)**

```
citește x (număr natural nenul)
cât timp x>0 execută
    citește y (număr natural)
    dacă x>y 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. Câte grafuri neorientate, distincte, cu 8 vârfuri se pot construi? Două grafuri se consideră distincte dacă matricele lor de adiacență sunt diferite. (4p.)
- a.  $4^{14}$                       b.  $2^{14}$                       c.  $4^{28}$                       d. 64
2. Variabila `t`, declarată alăturat, memorează în câmpurile `a`, `b` și `c` lungimile laturilor unui triunghi. Care dintre următoarele instrucțiuni atribuie câmpului `p` al variabilei `t` valoarea perimetrului triunghiului respectiv? (4p.)
- ```
struct triunghi
{
    float a,b,c,p;
}t;
```

a. `p.t=t.a+t.b+t.b;`                      b. `p.t=a.t+b.t+c.t;`

c. `t.p=t.a+t.b+t.c;`                      d. `t.p==t.a+t.b+t.c;`

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

3. Se consideră o stivă în care inițial au fost introduse, în această ordine, elementele cu valorile 1, 2 și 3. Se notează cu **AD(x)** operația prin care se adaugă elementul cu valoarea `x` în vârful stivei și cu **EL** operația prin care se elimină elementul din vârful stivei. Asupra acestei stive se execută următoarea secvență de operații: **AD(4);EL;AD(5);EL;AD(6);EL;EL**.
- a) Care este valoarea elementului din vârful stivei în urma executării acestei secvențe de operații? (3p.)
- b) Care este suma valorilor elementelor aflate în stivă în urma executării acestei secvențe de operații? (3p.)
4. În secvența de program alăturată, variabila `a` memorează o matrice cu `n` linii și `n` coloane (numerotate de la 0 la `n-1`) cu elemente numere **întregi**, iar toate celelalte variabile sunt întregi. Știind că `n` este un număr natural **par**, nenul, scrieți instrucțiunile care pot înlocui punctele de suspensie din secvența de program alăturată astfel încât, în urma executării acesteia, să se afișeze ultima cifră a sumei elementelor **pozitive** de pe linia `k` ( $0 \leq k < n$ ) a matricei `a`. (6p.)
- ```
s = 0;
for(j = 0; j < n/2; j++)
{
    .....
}
printf("%d",s); | cout<<s;
```
5. Să se scrie un program C/C++ care citește de la tastatură un cuvânt format din cel mult 20 de caractere, doar litere mici ale alfabetului englez. Programul determină transformarea cuvântului citit prin înlocuirea fiecărei vocale a cuvântului, cu un șir format din două caractere și anume vocala respectivă urmată de litera mare corespunzătoare, restul literelor nemodificându-se, ca în exemplu. Programul afișează pe ecran cuvântul obținut, pe o singură linie. Se consideră vocale literele din mulțimea {a,e,i,o,u}.

**Exemplu:** pentru cuvântul bacalaureat se va afișa pe ecran: bAaCaAlaAuUreEaAt  
(10p.)

**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 ultimul cuvânt generat? (4p.)
- a. edcb                      b. eeee                      c. edde                      d. eded

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

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(12345);?** (6p.)

<pre>//C void f(long n) { printf("%d",n%10);   if(n!=0)     { f(n/100); printf("%d",n%10); } }</pre>	<pre>//C++ void f(long n) { cout&lt;&lt;n%10;   if(n!=0)     { f(n/100); cout&lt;&lt;n%10; } }</pre>
--	--

3. Fișierul text **NR.TXT** conține pe o singură linie, separate prin câte un spațiu, cel mult 100 de numere **întregi**, fiecare număr având cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele **naturale nenule** din fișier. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul **NR.TXT** conține numerele: -3 -10 0 7 -5 7 51 -800 6 3798, atunci pe ecran se va afișa: 6 7 7 51 3798 (10p.)

4. Un număr **n** se numește **extraprim** dacă atât el, cât și orice număr obținut prin permutarea cifrelor lui **n**, sunt numere prime. De exemplu, numărul 113 este un număr **extraprim** deoarece 113, 311, 131 sunt numere prime.

**a)** Scrieți definiția completă a unui subprogram **f**, cu un parametru, subprogram care:

- primește prin intermediul parametrului **a** un număr natural cu cel mult 3 cifre ( $a > 1$ )
- returnează suma tuturor exponenților din descompunerea în factori primi a valorii parametrului **a**.

**Exemplu:** pentru  $a=90$  subprogramul va returna valoarea 4, deoarece  $a=2 \cdot 3^2 \cdot 5$  și  $1+2+1=4$ . (4p.)

**b)** Scrieți un program C/C++ care citește de la tastatură un număr natural **n**,  $2 \leq n \leq 999$  și, folosind apeluri utile ale subprogramului **f**, verifică dacă **n** este un număr **extraprim**. În caz afirmativ, programul afișează pe ecran mesajul **DA**, în caz contrar afișând mesajul **NU**. (6p.)