

EXAMENUL DE BACALAUREAT – 2007
Proba scrisă la INFORMATICĂ
PROBA E, limbajul C/C++

Varianta 30

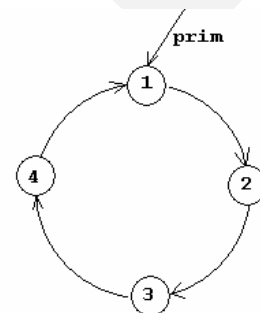
- ♦ Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- ♦ Timpul efectiv de lucru este de 3 ore.
- ♦ În programele cerute la subiectele II și III, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

SUBIECTUL I (40 de puncte)

Pentru fiecare din itemii de la 1 la 8, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 5 puncte.

1. Se consideră lista circulară simplu înlănțuită din figura alăturată în care fiecare element memorează în câmpul **nr** un număr natural și în câmpul **urm** adresa elementului următor din listă. Pentru variabila **prim** din figură, stabiliți câte **treceri** sunt necesare pentru ca toate elementele din listă să ajungă egale. Definim prin **trecere** prelucrarea dată de secvența următoare :

```
p=prim;
do{ if(p->nr < p->urm->nr) p->nr = p->nr +1;
    p=p->urm;
} while(p!=prim);
```



- a. 5 b. 2 c. 3 d. 4
2. Construim anagramele unui cuvânt $L_1L_2L_3$ prin generarea în ordine lexicografică a permutărilor indicilor literelor cuvântului: $L_1L_2L_3, L_1L_3L_2, L_2L_1L_3, L_2L_3L_1, L_3L_1L_2, L_3L_2L_1$. Pentru anagramele cuvântului **dac**, după șirul **dac, dca, adc, acd**, cuvintele imediat următoare sunt, în ordine:
- a. cda, dca b. cad, cda c. adc, cad d. cda, cad
3. Condiția ca numărul natural **m** să fie multiplu al numerelor naturale **a** și **b** este:
- a. $a \% m == 0 \ || \ b \% m == 0$ b. $a \% m == 0 \ \&\& \ b \% m == 0$
c. $m \% a == 0 \ || \ m \% b == 0$ d. $m \% a == 0 \ \&\& \ m \% b == 0$
4. Pentru reprezentarea unui arbore cu 8 noduri, numerotate cu numere de la 1 la 8, se utilizează vectorul de tați **TATA** = (3, 4, 7, 7, 4, 7, 0, 5). Care sunt frunzele arborelui?
- a. 1, 2, 3, 8 b. 3, 4, 5, 7 c. 1, 2, 6, 8 d. 1, 2, 3, 4
5. Niciunul dintre numerele reale **x** și **y** nu aparține intervalului **[a, b]** dacă și numai dacă:
- a. $(x < a \ || \ x > b) \ \&\& \ (y < a \ || \ y > b)$ b. $x < a \ \&\& \ y < a \ || \ x > b \ \&\& \ y > b$
c. $x < a \ || \ x > b \ \&\& \ y < a \ || \ y > b$ d. $x * y < a * a \ || \ x * y > b * b$
6. Graful orientat $G=(X, U)$ are 20 de vârfuri numerotate de la 1 la 20 și arce între vârfurile numerotate **i** și **j** care îndeplinesc condițiile: **i** este număr de o singură cifră iar **j** este un număr de două cifre ce are în scrierea sa cifra **i**. Numărul valorilor de 1 din matricea de adiacență asociată grafului **G** este:
- a. 20 b. 19 c. 10 d. 15
7. Pentru definiția subprogramului alăturat stabiliți ce se afișează la apelul **f(5, 1)**.
- ```
void f(int n, int k)
{if(k<=n){
 printf("%d",n-k); | cout<<n-k;
 f(n,k+1);
}}
```
- a. 12345                                      b. 01234                                      c. 43210                                      d. 54321
8. Pentru a verifica dacă toate elementele unui vector ordonat descrescător **A** sunt strict mai mici decât toate elementele unui alt vector **B** ordonat crescător, se compară primul element din **A** cu primul element din **B**. Această metodă de verificare este:
- a. corectă numai pentru componente întregi                              b. corectă și neeficientă  
c. corectă și eficientă                                      d. incorectă

**SUBIECTUL II (20 de puncte)**

Se consideră programul pseudocod alăturat în care s-a folosit notația  $[a]$  pentru partea întreagă a numărului real  $a$ .

1. Care este valoarea afișată pentru  $n=1234$ ? (6p.)
2. Scrieți o valoare de două cifre pentru variabila  $n$  astfel încât să se afișeze 1. (2p.)
3. Pentru câte valori distincte ale lui  $n$ , număr natural cu maximum 3 cifre se afișează valoarea 0? (2p.)
4. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

```

citește n (număr întreg, n>0)
k ← 0
c ← 0
naux ← n
┌cat timp naux>0 execută
│naux ← [naux/10]
│k ← k+1
│c ← c*10+1
└─
┌pentru i ← 1, k execută
│n ← n-c
│c ← [c/10]
└─
scrie n

```

**SUBIECTUL III (30 de puncte)**

1. Pentru o valoare  $n$  (număr natural,  $1 \leq n \leq 100$ ) citită de la tastatură scrieți programul C/C++ care scrie în fișierul `bac.txt` un tablou bidimensional cu  $n$  linii și  $n$  coloane cu formatul alăturat. Elementele de pe fiecare linie sunt separate prin spațiu.  
De exemplu, pentru  $n=4$ , conținutul fișierului `bac.txt` este:

```

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

```

(10p.)

```

1 0 0 0 ...0
2 1 0 0 ...0
3 2 1 0 ...0
.....
n n-1 n-2 n-3 ...1

```

2. a) Scrieți **numai** antetul subprogramului `divizor`, care primește prin intermediul parametrului  $n$  un număr natural ( $n>1$ ) cu maximum 9 cifre și returnează prin intermediul parametrului  $d$  valoarea celui mai mic divizor prim al lui  $n$ , iar prin intermediul parametrului  $p$  puterea la care acest divizor apare în descompunerea în factori primi a numărului  $n$ . (2p.)  
b) Scrieți programul C/C++ care citește de la tastatură două numere naturale  $n, x$  ( $x, n>1$ ) cu maximum 9 cifre și verifică dacă  $n$  este divizibil cu  $2^x$ , folosind apeluri ale funcției `divizor` definită la punctul a). Programul afișează **DA** în caz afirmativ și **NU** în caz contrar. (8p.)
3. Se citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 100$ ). Să se afișeze pe ecran al  $n$ -lea termen al șirului 11, 22, 33, 44, 55, 66, 77, 88, 99, 111, 222, 333, 444, etc.  
De exemplu, dacă  $n=11$  se afișează 222.

- a) Alegeți o metodă eficientă de rezolvare, descriind în limbaj natural metoda folosită și justificați eficiența acesteia (cel mult 6 rânduri). (2p.)
- b) Scrieți programul C/C++ corespunzător metodei descrise la punctul a). (8p.)