

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

Varianta 69

- ♦ 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).

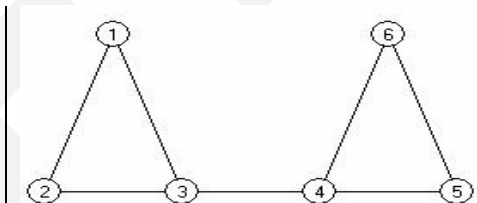
**SUBIECTUL I (40 de puncte)**

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. Care trebuie să fie valoarea inițială a variabilei *i* de tip întreg pentru ca în urma executării instrucțiunii alăturate, pe ecran să fie afișată secvența de caractere \*\*\*\*\* ?
 

```
while (i*5<1000)
{
    printf(","); / cout<<" ";
    i=i*2+10;
}
```

a. 3                      b. 11                      c. 13                      d. 5
2. Se consideră graful neorientat din figura alăturată:  
 Care este numărul cel mai mic de muchii care trebuie adăugate pentru ca graful să devină eulerian ?
 



a. 3                      b. 2                      c. 4                      d. 1
3. Se consideră subprogramul recursiv cu definiția alăturată :
 

```
void p(int n)
{ if (n!=1)
  { printf("%d ",n); / cout<<n<<" ";
    if(n%2==0) p(n/2);
    else p(3*n+1);
  }
  else printf("%d",1); / cout<<1;
}
```

Ce valori vor fi afișate pe ecran în urma apelului *p*(10) ?

a. 5 16 8 4 2 1                      b. 10 5 16 8 4 2 1

c. 10 5 16 8 4 2                      d. 10 5 4 2 1
4. Se consideră graful orientat  $G = (X, U)$  unde  $X = \{1, 2, 3, 4, 5, 6\}$  și  $U = \{(1,2), (1,5), (1,6), (2,3), (3,5), (4,1), (5,4)\}$ . Identificați care sunt nodurile accesibile din toate celelalte noduri ale grafului prin intermediul unor drumuri elementare.
 

a. 6                      b. 1 5                      c. 1 2 3 5                      d. 4 5
5. Ce valoare va reține variabila *x* după executarea următoarei secvențe de atribuiri ?
 

```
x<-10;      y<-3;      x<-x-y;      y<-x+y;      x<-y-x
```

a. 10                      b. -3                      c. 7                      d. 3
6. Care dintre următorii vectori "de tați" corespunde reprezentării unui arbore în care nodurile numerotate cu 6, 4 și 9 sunt descendenți direcți ai nodului 3 ?
 

a. tata=(3, 3, 4, 0, 2, 3, 4, 4, 4)                      b. tata=(9, 9, 4, 9, 9, 9, 9, 9, 0)

c. tata=(3, 3, 1, 3, 2, 3, 4, 4, 3)                      d. tata=(3, 0, 2, 3, 2, 3, 4, 4, 3)
7. Știind că variabilele *a*, *b* și *c* sunt de tip întreg, care este condiția ca numărul natural memorat de variabila *c* să fie un multiplu comun al numerelor naturale memorate de variabilele *a* și *b* ?
 

a. *c*%*a* \* *c*%*b* == 0                      b. *a*%*c* + *b*%*c* == 0

c. *c*%*a* + *c*%*b* == 0                      d. (*a*%*c* == 0) && (*c*%*b* == 0)
8. Construim anagramele unui cuvânt  $L_1L_2L_3L_4$  prin generarea în ordine lexicografică a permutărilor indicilor literelor cuvântului și obținem  $L_1L_2L_3L_4$   $L_1L_2L_4L_3$   $L_1L_3L_2L_4$  ...  $L_4L_3L_1L_2$   $L_4L_3L_2L_1$ . Pentru anagramele cuvântului *caiet*, după șirul *caeit*, *caeti*, *catie* cuvintele imediat următoare sunt:
 

a. *catei* și *ciaet*                      b. *ciaet* și *caite*

c. *catei* și *ciate*                      d. *ciaet* și *ciate*

**SUBIECTUL II (20 de puncte)**

Se consideră programul pseudocod alăturat în care s-a notat cu  $x \div y$  restul împărțirii numărului natural  $x$  la numărul natural  $y$  și cu  $[z]$  partea întreagă a numărului real  $z$ .

1. Care este valoare afișată pentru  $n=52381$ ? (3p.)
2. Scrieți o valoare pentru variabila  $n$  astfel încât să se afișeze valoarea 0. (3p.)
3. Scrieți un program pseudocod echivalent cu algoritmul dat care să conțină o structură repetitivă cu test final. (6p.)
4. Scrieți programul C/C++ corespunzător algoritmului dat. (8 p.)

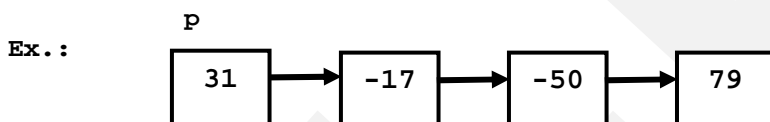
```

citește n {n număr natural}
z ← 0
cât timp n > 0 execută
    c ← n%10; n ← [n/10]
    dacă c%2 = 0
        atunci z ← z*10 + c
scrie z

```

**SUBIECTUL III (30 de puncte)**

1. Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $0 < n < 1000$ ). Să se afișeze pe ecran toate numerele naturale perfecte mai mici decât  $n$  separate printr-un spațiu. Un număr natural se numește număr perfect dacă este egal cu suma divizorilor săi, divizori din care se exclude divizorul egal cu numărul însuși ( $6 = 1 + 2 + 3$ ) (10p.)  
**Ex. :** pentru  $n=50$  se va afișa **6 28**
2. Se consideră o listă simplu înlănțuită (cu cel puțin două elemente) în care fiecare element reține în câmpul **info** un număr real nenul, iar în câmpul **adr** adresa următorului element din listă. Scrieți definițiile tipurilor de date și definiția completă a subprogramului **s1** care are ca parametru adresa **p** a primului element al listei și care modifică lista prin inserarea între oricare două elemente de semne contrare a unui nou element a cărui valoare este egală cu media aritmetică a celor două elemente. (10p.)



Se obține :



3. Se consideră un șir de cel mult 100 de litere mici ale alfabetului englez. Acest șir este supus unui proces de eliminare, la fiecare etapă eliminându-se toate secvențele formate din caractere identice situate pe poziții alăturate. Șirul nou obținut este supus aceluiași proces de eliminare până când nu mai există în șir caractere alăturate egale. Scrieți programul C/C++ care citește de la tastatură șirul și afișează în fișierul **DATE.TXT** șirul obținut după încheierea întregului proces de eliminare. (10 p.)

**Exemplu:**

Pentru șirul **teuuueusppi** după prima aplicare a procesului de eliminare se obține șirul **teeusi**, iar apoi șirul **tusi**. Programul va afișa în fișierul **DATE.TXT** numai șirul **tusi**, fără spații între literele ce-l formează.