

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

Varianta 53

- ♦ 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 un graf neorientat cu 10 vârfuri cu proprietatea că există muchie de la vârful i la vârful j dacă și numai dacă i și j sunt numere prime (numărul 1 se consideră că nu este prim). Care este numărul muchiilor din acest graf?
 - a. 7
 - b. 6
 - c. 9
 - d. 12
2. Care este numărul minim de muchii ce trebuie eliminate astfel încât graful neorientat cu 6 noduri și cu matricea de adiacență alăturată să fie eulerian?

0	1	1	1	1	0
1	0	0	1	1	1
1	0	0	1	0	0
1	1	1	0	1	0
1	1	0	1	0	1
0	1	0	0	1	0

 - a. 4
 - b. 1
 - c. 0
 - d. 2
3. Se consideră următorul subprogram recursiv:


```
void p(int i,int n)
{int j;
  if(i<=n)
  {p(i+1,n);
   for(j=1;j<=i;j++)
     cout<<j; | printf("%d",j);
   cout<<endl; | printf("\n");
  }
}
```

 Ce se va afișa în urma apelului $p(1,3)$?
 - a. 123
23
3
 - b. 123
12
1
 - c. 321
32
3
 - d. 1
12
123
4. Câte numere de 10 cifre pot fi obținute utilizând numai cifrele 0 și 9?
 - a. 2^{10}
 - b. 2^9
 - c. 9
 - d. 10
5. Care este numărul grafurilor orientate cu n noduri cu proprietatea că pentru orice pereche de noduri distincte i și j există cel puțin un arc între i și j .
 - a. 3^n
 - b. $n!$
 - c. 2^n
 - d. $3^{n*(n-1)/2}$
6. Valoarea expresiei următoare $(x > -2) \&\& (! (x > 2)) \mid (x \geq 5) \&\& (x < 10)$ este diferită de 0 dacă și numai dacă x aparține intervalului:
 - a. $(-2, 2] \cup [5, 10)$
 - b. $[-2, 2] \cup [5, 10)$
 - c. $(2, 5]$
 - d. $(-2, 10)$
7. Se consideră următoarele declarații:


```
int x=3,i=0;
```

 Ce va afișa secvența alăturată?


```
while(x-1){x--;i++;}
cout<<i; | printf("%d",i);
```

 - a. 1
 - b. 0
 - c. 2
 - d. 4
8. Considerăm următoarea declarație:


```
struct produs{char denumire[10];int pret;} p;
```

 Cum se poate accesa prima literă a denumirii unui produs ale cărui caracteristici sunt memorate în variabila p ?
 - a. $p.denumire[1]$
 - b. $p->denumire$
 - c. $p.denumire[0]$
 - d. $p->denumire[1]$

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat.

S-au folosit următoarele notații: **mod** pentru restul împărțirii întregi iar **div** pentru câtul împărțirii întregi.

1. Ce se va afișa pentru $x=1232189$? (5p.)
2. Dați o valoare pentru numărul x astfel încât algoritmul să nu afișeze nimic. (3p.)
3. Scrieți un program pseudocod care să fie echivalent cu cel dat, dar în care să se înlocuiască structura repetitivă **pentru** cu o structură repetitivă cu test inițial. (4p.)
4. Scrieți programul C/C++ corespunzător algoritmului dat. (8p.)

```

citește x {x nr.natural}
pentru c ← 0, 9 execută
    y ← x
    p ← 0
    repetă
        dacă y mod 10 = c atunci
            p ← p + 1
        y ← y div 10
    până când y = 0
    dacă p > 1 atunci
        scrie c

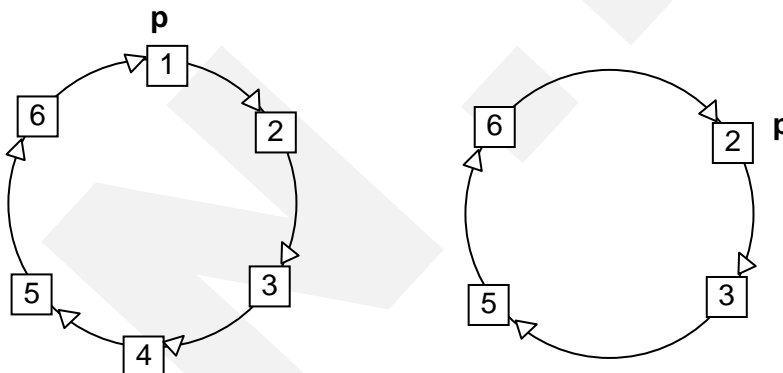
```

SUBIECTUL III (30 de puncte)

1. Se consideră o listă circulară simplu înlănțuită, fiecare nod al listei reținând în câmpul **leg** adresa elementului următor al listei. Componentele listei memorează în câmpul **info**, în ordine, numere naturale consecutive începând cu numărul 1.

Scrieți declarațiile de date și un subprogram **elimin** ce realizează ștergerea din listă a nodurilor ce memorează numere pătrate perfecte. Subprogramul **elimin** are un singur parametru p , reprezentând adresa nodului ce memorează valoarea 1.

În exemplul ilustrat de figura următoare, dacă lista conține inițial numerele de la 1 la 6, după apelul subprogramului **elimin(p)** aceasta va conține, în ordine, valorile 2, 3, 5, 6.



(10p.)

2. Prin secțiune a unui șir v înțelegem o succesiune de elemente ale lui v situate pe poziții consecutive în șirul dat. Dacă în fișierul text **bac.in** se află pe prima linie un număr natural n ($1 < n < 10000$) iar pe linia următoare cele n componente întregi ale șirului v , componente cu cel mult patru cifre fiecare, se cere să se determine în mod eficient secțiunea de sumă minimă. Se va afișa pe ecran numărul reprezentând suma minimă determinată.

a) Explicați în limbaj natural metoda utilizată, justificând eficiența acesteia (4-5 rânduri). (2p.)

b) Scrieți programul **C/C++** corespunzător metodei descrise la punctul a). (8p.)

De exemplu, dacă fișierul **bac.in** conține:

```

6
-3 2 -3 -4 5 6

```

se va afișa :

-8

3. Se citesc de la tastatură două numere naturale n și p ($1 < n < 1000, 1 < p < 10$). Să se afișeze pe ecran, cu spațiu între ele, acele numere naturale mai mici sau egale cu n care au toate cifrele mai mici sau egale cu p .

De exemplu, dacă $n=15$ și $p=2$, se vor afișa :

0 1 2 10 11 12

(10p.)