

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

Varianta 76

- ♦ 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. Variabila întregă **nr** trebuie să rețină câți multipli mai mari decât 0 și mai mici sau egali decât **n** are numărul **k** (**n** și **k** sunt numere naturale date). Care este expresia cu care trebuie completată atribuirea **nr=...**?
 - a. **n%k**
 - b. **n-n/k**
 - c. **n/k**
 - d. **n-n%k**
2. Se consideră un graf neorientat cu 9 noduri și muchiile [1,2], [4,8], [5,9], [2,3], [7,8], [3,7], [6,9], [6,7], [4,6], [4,5], [1,7]. Numărul minim de muchii care trebuie adăugate pentru ca graful să devină eulerian este
 - a. 5
 - b. 0
 - c. 25
 - d. 2
3. Se consideră funcția alăturată. Ce se va afișa pe ecran la apelul **f(13)**?


```
void f(int x)
{
    if (x>1) f(x/2);
    printf("%d",x%2); | cout<<x%2;
}
```

 - a. 0101
 - b. 1100
 - c. 1101
 - d. 1011
4. Într-o listă simplu înălțuită fiecare element reține în câmpul **adr** adresa elementului următor din listă, iar în câmpul **inf** un număr întreg. Adresa primului element al listei este memorat în variabila **p**. Dacă în listă sunt memorate, în această ordine, numerele

p
 ↳ [2] → [6] → [9] → [4] → [5] → [3]

 Știind că variabila **c** este de același tip cu **inf**, în urma executării secvenței de instrucțiuni:
c=p->adr->inf; p->adr->inf=p->adr->adr->inf; p->adr->adr->inf=c;
 în listă vor fi memorate în ordine numerele:
 - a. 2 6 9 4 5 3
 - b. 6 9 4 5 3 2
 - c. 2 9 6 4 5 3
 - d. 6 2 9 4 5 3
5. Dacă se utilizează metoda backtracking pentru a genera toate permutările de 4 obiecte și primele 5 permutări generate sunt: 4 3 2 1, 4 3 1 2, 4 2 3 1, 4 2 1 3, 4 1 3 2, atunci a 6-a permutare este:
 - a. 3 4 2 1
 - b. 4 1 2 3
 - c. 3 2 1 4
 - d. 1 4 3 2
6. În subprogramul alăturat, **y** este:


```
int f(float x)
{int y;
    if(x>0)y=1; else y=-1;
    return y;
}
```

 - a. variabilă locală
 - b. variabilă globală
 - c. parametru formal
 - d. numele funcției
7. Se consideră arborele cu 8 noduri și muchiile [1,5], [2,3], [3,6], [3,8], [4,6], [5,7], [6,7]. Care dintre nodurile arborelui ar putea fi alese ca rădăcină pentru ca arborele să aibă număr maxim de niveluri:
 - a. 1,2,8
 - b. 3,4,7
 - c. 6
 - d. 5
8. Stabiliți care dintre următoarele expresii este adevărată dacă și numai dacă **x** este în afara intervalului închis [a,b]
 - a. **x<a && x>b**
 - b. **x>=a && x<=b**
 - c. **!(x>=a && x<=b)**
 - d. **!(x<a || x>b)**

SUBIECTUL II (20 de puncte)

Se consideră programul pseudocod alăturat:

S-a notat cu $x|y$ relația „ x divide pe y ” sau „ y este divizibil cu x ” și cu $[x]$ partea întreagă a numărului real x .

1. Ce se va afișa pentru $n=112$? (5 p.)
2. Scrieți o valoare pentru variabila n astfel încât să se afișeze o valoare egală cu n . (5 p.)
3. Scrieți programul C/C++ corespunzător algoritmului dat. (10 p.)

```

citește n
i ← 2
p ← 1
cât timp n > 1
    k ← 0
    cât timp i | n
        k ← i
        n ← [n/i]
    dacă k ≠ 0 atunci p ← p * k
    i ← i + 1
scrie p

```

SUBIECTUL III (30 de puncte)

1. Să se scrie programul C/C++ care citește de la tastatură un cuvânt de maximum 20 de litere și minimum o literă și șterge litera din mijloc în cazul în care cuvântul are un număr impar de litere, respectiv cele două litere din mijlocul cuvântului dacă acesta are un număr par de litere. Programul va afișa cuvântul rezultat în urma ștergerii sau mesajul CUVANT VID dacă după ștergere acesta rămâne fără nici o literă.

Exemplu : dacă se citește cuvântul **carte**, se va afișa **cate**.

(10p.)

2. Pe prima linie a fișierului **BAC.TXT** se găsesc numere naturale separate prin câte un spațiu. Ele formează mai multe șiruri crescătoare de numere naturale, sfârșitul fiecărui șir fiind marcat de valoarea -1 (care nu face parte din nici un șir). Știind că valorile numerelor din șiruri nu depășesc, în valoare absolută, 10000 și că în total sunt cel mult 2000 de numere, scrieți un program care să afișeze numerele comune tuturor șirurilor, în ordinea crescătoare a valorilor lor. De exemplu, dacă fișierul conține:

2 3 3 4 5 8 9 -1 2 4 5 8 11 -1 2 3 5 8 12 -1

atunci se va afișa: 2 5 8

a) Descrieți în limbaj natural o metodă eficientă de rezolvare ca timp de executare.

(2p.)

b) Scrieți programul C/C++ corespunzător metodei descrise la punctul a).

(8p.)

3. Scrieți programul C/C++ care construiește în memorie o matrice pătrată cu n linii și n coloane formată numai din valori cuprinse între 1 și n astfel încât să nu existe două elemente egale pe aceeași linie și nici pe aceeași coloană. Prima linie trebuie să conțină, în această ordine, numerele 1, 2, ..., n ; linia a doua va avea, în ordine, numerele 2, 3, ..., n , 1; linia a treia va avea, în ordine, numerele 3, 4, ..., n , 1, 2, iar pe ultima linie numerele n , 1, 2, 3, ..., $n-1$. Valoarea lui n (număr natural, $2 < n < 25$) se citește de la tastatură, iar matricea construită se afișează, pe linii, pe ecran. De exemplu, pentru $n=4$ se va afișa:

```

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

```

(10p.)