

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

Varianta 65

- ♦ 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 folosește o metodă de generare a tuturor partițiilor mulțimii  $A = \{1, 2, 3\}$  obținându-se următoarele soluții:  $\{1\}\{2\}\{3\}; \{1\}\{2, 3\}; \{1, 3\}\{2\}; \{1, 2\}\{3\}; \{1, 2, 3\}$ . Se observă că dintre acestea, prima soluție e alcătuită din exact trei submulțimi. Dacă se folosește aceeași metodă pentru a genera partițiile mulțimii  $\{1, 2, 3, 4\}$  stabiliți câte dintre soluțiile generate vor fi alcătuite din exact trei submulțimi.
  - a. 3
  - b. 6
  - c. 12
  - d. 5
2. Subprogramul `divizori(x,y)` returnează numărul de divizori comuni a numerelor naturale  $x$  și  $y$ .  
 Frația  $\frac{a}{b}$ , cu  $a, b$  numere naturale și  $b \neq 0$ , este ireductibilă dacă este satisfăcută condiția:
  - a. `divizori(a,b) == 0`
  - b. `divizori(a,b) / divizori(b,a) != 1`
  - c. `divizori(a,divizori(b)) == 0`
  - d. `divizori(a,b) == 1`
3. Care dintre următoarele secvențe de instrucțiuni determină afișarea pe ecran, în urma executării, a numărului 55 ( $i$  și  $j$  fiind variabile de tip `int`)?
  - a. `i=5; j=6; while(j>4) printf("%d",j);/cout<<j; j--;`
  - b. `i=5; j=6; while(j>4){printf("%d",i);/cout<<i; j--;}`
  - c. `j=5; for(i=5;i<=5;i++) printf("%d",i);/cout<<i;`
  - d. `j=5; for(i=1;i<2;i++) printf("%d",j);/cout<<j;`
4. Într-un graf neorientat  $G$ , notăm cu  $n$  numărul de vârfuri și cu  $m$  numărul de muchii. Dacă graful este un arbore atunci între  $n$  și  $m$  există următoarea relație matematică:
  - a.  $m=n+2$
  - b.  $n=m-1$
  - c.  $n=m+1$
  - d.  $n=m+2$
5. Graful orientat  $G$  cu 10 noduri, reprezentat prin listele de adiacență alăturate, are:
 

<code>1: 4 6</code> <code>2: 1</code> <code>3:</code> <code>4: 6</code> <code>5: 7 9</code>	<code>6:</code> <code>7:</code> <code>8:</code> <code>9: 8</code> <code>10:</code>
---	--

  - a. Două drumuri distincte de la nodul 2 la nodul 6
  - b. Un drum de la nodul 7 la nodul 8
  - c. Un circuit care conține nodurile 1, 4, 6
  - d. Două drumuri distincte de la nodul 5 la nodul 8
6. Variabila `b` reține în câmpul `fraza` un text, format din cel mult 255 de caractere și în câmpul `nr1` numărul de litere mici din text. Care dintre referirile următoare reprezintă primul caracter din câmpul `fraza` al variabilei `b`?
 

- a. `b.fraza[0]`
  - b. `b.fraza`
  - c. `b[0].fraza[0]`
  - d. `b[0].fraza`

```
struct
{char fraza[256],int nr1;} b;
```
7. De câte ori se apelează funcția `f` în timpul executării atribuirii `x=f(f(999))`, cu `x` întreg?
 

- a. 4
  - b. 6
  - c. 2
  - d. 5

```
int f(int n)
{if (n==0) return 0;
 else return 1+f(n/10);}
```
8. Subprogramul `max(n)` returnează cea mai mare cifră a numărului natural  $n$ . Pentru  $n$  număr natural, format din 3 cifre, expresia `max(n/10%10)+max(n%10)+max(n/100)` reprezintă:
  - a. Cifra unităților numărului  $n$
  - b. Cifra maximă a numărului  $n$
  - c. Numărul cifrelor numărului  $n$
  - d. Suma cifrelor numărului  $n$

**SUBIECTUL II (20 de puncte)**

Se consideră programul pseudocod alăturat:

1. Care sunt valorile afișate în urma executării, dacă se citește succesiunea de valori: 5, 7, 8, 1, 0, 6 ? (5 p.)
2. Precizați o succesiune de 7 valori care pot fi citite astfel încât instrucțiunile din structura repetitivă **cât timp** să nu se execute niciodată. (3p.)
3. Scrieți programul C/C++ corespunzător algoritmului dat. (6p.)
4. Scrieți un program pseudocod echivalent cu algoritmul dat, care utilizează o structură repetitivă cu test final în locul structurii repetitive **cât timp**. (6p.)

```

citește n { n număr natural,
1 < n ≤ 100 }

pentru i ← 1, n execută
    citește ai
    ■

pentru i ← 2, n execută
    x ← ai; j ← i - 1
    cât timp j > 0 și x < aj execută
        aj+1 ← aj
        aj ← x
        j ← j - 1
    ■
    ■

pentru i ← 1, n execută
    scrie ai
    ■
  
```

**SUBIECTUL III (30 de puncte)**

1. Scrieți programul C/C++ care citește de la tastatură un număr natural  $n$  ( $n < 100$ ) și un șir cu  $n$  numere întregi din intervalul  $[100, 999]$ ; programul construiește în mod eficient din punctul de vedere al spațiului de memorie folosit, un șir de numere rezultat prin înlocuirea fiecărui număr din șirul citit cu numărul obținut prin interschimbarea cifrei unităților cu cifra sutelor. Numerele din noul șir se vor afișa pe ecran separate printr-un singur spațiu. De exemplu, pentru  $n=3$  și șirul 123 904 500, se afișează: 321 409 5. (10p.)

2. Spunem că secvența de  $k$  numere  $x_1, x_2, \dots, x_k$  este mai mică în ordine lexicografică decât secvența de  $k$  numere  $y_1, y_2, \dots, y_k$  dacă există o poziție  $i$  ( $1 \leq i \leq k$ ) astfel încât  $x_1 = y_1, x_2 = y_2, \dots, x_{i-1} = y_{i-1}$  și  $x_i < y_i$ .

Scrieți definiția completă a unui subprogram în limbajul C/ C++ care primește:

- prin intermediul parametrilor  $a$  și  $b$  două tablouri unidimensionale cu același număr de elemente de tip `int`;
- prin intermediul parametrului  $n$  ( $n$  număr natural,  $1 < n \leq 1000$ ) numărul de elemente pe care îl are fiecare dintre cele două tablouri.

Subprogramul returnează valoarea 1 dacă  $a$  este mai mic în ordine lexicografică decât  $b$  și 0 în caz contrar. (10p.)

3. Un șir crescător de fracții ireductibile din intervalul  $[0, 1]$ , cu numitorul mai mic sau egal cu  $n$ , se numește șir Farey de ordin  $n$ , notat  $F_n$ . Șirurile Farey de ordin 1, 2 și 5 conțin elementele:

$$F_1 = \left\{ \frac{0}{1}, \frac{1}{1} \right\}; \quad F_2 = \left\{ \frac{0}{1}, \frac{1}{2}, \frac{1}{1} \right\}, \dots, \quad F_5 = \left\{ \frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{4}{5}, \frac{1}{1} \right\}$$

Dacă notăm o astfel de serie de valori cu:  $x_0/y_0, x_1/y_1, \dots, x_i/y_i, \dots$  atunci valorile care apar într-un șir Farey se pot calcula cu următoarele relații:

$$x_0=0, y_0=1, x_1=1, y_1=n, \quad x_{i+2} = [(y_i+n)/y_{i+1}]x_{i+1} - x_i, \quad y_{i+2} = [(y_i+n)/y_{i+1}]y_{i+1} - y_i,$$

în care s-a notat cu  $[a]$  partea întreagă a lui  $a$ .

Scrieți un program C/C++ care citește de la tastatură un număr natural  $n$  ( $1 \leq n \leq 50$ ) și apoi creează fișierul text `numere.txt` care conține pe fiecare linie, separate prin spațiu, numărătorul și numitorul unei fracții din șirul Farey de ordinul  $n$ .

De exemplu pentru  $n=2$ , conținutul fișierului `numere.txt` va fi:

```

0 1
1 2
1 1
  
```

(10p.)