

Examenul de bacalaureat 2011
Proba E. d)
Proba scrisă la INFORMATICĂ
Limbajul C/C++

Varianta 9

Filiera teoretică, profilul real, specializările: **matematică – informatică**
matematică – informatică intensiv informatică
Filiera vocațională, profilul militar, specializarea **matematică – informatică**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, 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 **(30 de puncte)**

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Indicați care dintre expresiile C/C++ de mai jos are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x aparține reuniunii de intervale $[-3, -1] \cup [1, 3]$. **(4p.)**
- $x >= -3 \ \&\& \ x <= -1 \ \&\& \ x >= 1 \ \&\& \ x <= 3$
 - $!(x < -3 \ || \ x > -1) \ || \ !(x < 1 \ || \ x > 3)$
 - $x >= -3 \ || \ x <= -1 \ || \ x >= 1 \ || \ x <= 3$
 - $!(x < -3 \ \&\& \ x > 3 \ \&\& \ x > -1 \ || \ x < 1)$

2. Se consideră algoritmul alăturat, descris în pseudocod.

- Scrieți numerele care se afișează în urma executării algoritmului, în ordine, dacă pentru n se citește valoarea 5, iar pentru k se citește valoarea 2. **(6p.)**
- Dacă pentru variabila k se citește valoarea 5, scrieți cea mai mică și cea mai mare valoare care pot fi citite pentru variabila n astfel încât, în urma executării algoritmului, ultimul număr care se afișează, pentru fiecare dintre aceste valori, să fie 7. **(6p.)**
- Scrieți în pseudocod un algoritm care să conțină o singură structură repetitivă în loc de două și care să fie echivalent cu cel dat. **(4p.)**
- Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

```
citește n, k
    (numere naturale nenule)
t ← 0
cât timp n ≥ 1 execută
    dacă n > k atunci i ← k
    altfel i ← n
    ■
    t ← t + 1
    n ← n - i
cât timp i ≥ 1 execută
    scrie t, ' '
    i ← i - 1
    ■
    ■
```

SUBIECTUL al II-lea

(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Numim pădure un graf neorientat în care fiecare componentă conexă a sa este un arbore. Orice pădure cu cel puțin doi arbori este un graf care: **(4p.)**
 - a. are cicluri și este conex
 - b. are cicluri și nu este conex
 - c. nu are cicluri și este conex
 - d. nu are cicluri și nu este conex
2. Se consideră graful orientat cu 6 vârfuri, numerotate de la 1 la 6, și mulțimea arcelor $\{(1, 4), (2, 1), (2, 4), (2, 6), (4, 1), (4, 6)\}$. Numărul de subgrafuri ale sale, fiecare având mulțimea arcelor $\{(1, 4), (2, 1), (2, 4), (4, 1)\}$, este: **(4p.)**
 - a. 2
 - b. 3
 - c. 4
 - d. 5

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Se consideră graful neorientat cu 5 noduri, numerotate de la 1 la 5, reprezentat prin listele de adiacență alăturate. Enumerați nodurile care fac parte din componenta conexă a grafului dat care are cele mai puține noduri. **(6p.)**

1:	4, 5
2:	3
3:	2
4:	1
5:	1
4. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg și variabila A este de tip tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4, cu elemente numere întregi. Fără a utiliza alte variabile, scrieți o instrucțiune care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. Înainte de executarea secvenței toate elementele tabloului sunt nule. **(6p.)**

<pre>for (i=0; i<5; i++) for (j=0; j<5; j++) </pre>	<table style="border-collapse: collapse;"><tr><td style="border-right: 1px solid black; padding-right: 10px;">0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td style="border-right: 1px solid black; padding-right: 10px;">1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td style="border-right: 1px solid black; padding-right: 10px;">2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td style="border-right: 1px solid black; padding-right: 10px;">3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td style="border-right: 1px solid black; padding-right: 10px;">4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7	4	5	6	7	8
0	1	2	3	4																						
1	2	3	4	5																						
2	3	4	5	6																						
3	4	5	6	7																						
4	5	6	7	8																						
5. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($2 \leq n \leq 20$) și apoi n cuvinte distincte, fiecare fiind format din cel mult 20 de caractere, numai litere mici ale alfabetului englez. La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul afișează pe ecran numărul de cuvinte dintre ultimele $n-1$ citite, care încep cu primul cuvânt citit. **Exemplu:** dacă $n=5$ și cuvintele citite sunt:
bun
buncar
bunici
abundent
bunavoie
pe ecran se afișează 3 (deoarece numai cuvintele **buncar**, **bunici** și **bunavoie** încep cu **bun**). **(10p.)**

SUBIECTUL al III-lea

(30 de puncte)

Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră subprogramele recursive $c1$ și $c2$, definite mai jos.

<pre>int C1 (int a, int b) { if(a==b) return a; else if(a>b) return C1(a-b,b); else return C1(a,b-a); }</pre>	<pre>int C2 (int a, int b) { if(b==0) return a; else return C2(b,a%b); }</pre>
--	--

La apel, returnează valoarea celui mai mare divizor comun al celor două numere naturale nenule primite ca parametri: **(4p.)**

- a. numai $c1$ b. numai $c2$ c. atât $c1$ cât și $c2$ d. nici $c1$ nici $c2$

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

2. Având la dispoziție cinci mărgelile de culori diferite, **roșu, galben, verde, albastru, violet**, se utilizează metoda backtracking pentru a obține toate posibilitățile de a forma șiruri de câte trei mărgelile, știind că în cadrul unui șir contează ordinea de așezare a mărgelilor. Culorile mărgelilor, corespunzătoare primelor patru soluții obținute sunt, în această ordine: (**roșu, galben, verde**), (**roșu, galben, albastru**), (**roșu, galben, violet**), (**roșu, verde, galben**). Scrieți ultimele două soluții generate, în ordinea obținerii lor. **(6p.)**

3. Subprogramul **inserare** are doi parametri:

- n , prin care primește un număr natural ($2 \leq n \leq 20$);
- a , prin care primește un tablou unidimensional care memorează un șir de n numere naturale, fiecare cu cel mult 4 cifre. Cel puțin un element al tabloului este număr par.

Subprogramul modifică tabloul astfel încât după fiecare termen par al șirului inserează valoarea 2011 și furnizează, tot prin parametrii n și a , valorile actualizate ale datelor primite.

Scrieți în limbajul C/C++ definiția completă a subprogramului.

Exemplu: dacă $n=7$ și $a=(1, 4, 5, 3, 82, 6, 2)$ atunci, după apel, $n=11$ și $a=(1, 4, 2011, 5, 3, 82, 2011, 6, 2011, 2, 2011)$.

(10p.)

4. Se citesc de la tastatură două numere naturale $s1$ și $s2$ ($0 < s1 \leq 18$, $0 \leq s2 \leq 18$) și se cere scrierea în fișierul **BAC.TXT**, fiecare pe câte o linie, în ordine strict crescătoare, a tuturor numerelor naturale cu exact 5 cifre, pentru care suma primelor două cifre este egală cu $s1$, iar suma ultimelor două cifre este egală cu $s2$. Pentru determinarea numerelor indicate se utilizează un algoritm eficient din punct de vedere al timpului de executare.

Exemplu: dacă $s1=8$, iar $s2=7$, atunci 35725 este unul dintre numerele care respectă proprietatea cerută ($3+5=8$ și $2+5=7$).

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**