

**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 întreagă  $x$  aparține reuniunii de intervale  $[-3, -1] \cup [1, 3]$ . (4p.)
- a.  $x \geq -3 \ \&\& \ x \leq -1 \ \&\& \ x \geq 1 \ \&\& \ x \leq 3$
  - b.  $!(x < -3 \ || \ x > -1) \ || \ !(x < 1 \ || \ x > 3)$
  - c.  $x \geq -3 \ || \ x \leq -1 \ || \ x \geq 1 \ || \ x \leq 3$
  - d.  $!(x < -3 \ \&\& \ x > 3 \ \&\& \ x > -1 \ || \ x < 1)$

2. Se consideră algoritmul alăturat, descris în pseudocod.

- a. 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.)
- b. 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.)
- c. 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.)
- d. 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
```

**(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.
- ```
for (i=0; i<5; i++)  
    for (j=0; j<5; j++)  
        .....
```
- |   |   |   |   |   |
|---|---|---|---|---|
| 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 |
- (6p.)
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.

```
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);
}
```

```
int C2 (int a, int b)
{ if(b==0) return a;
  else return C2(b,a%b);
}
```

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.)**

**Examenul de bacalaureat 2011**  
**Proba E. d)**  
**Proba scrisă la INFORMATICĂ**

**BAREM DE EVALUARE ȘI DE NOTARE**  
**(comun pentru limbajele C/C++ și Pascal)**

**Varianta 9**

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

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct.
- Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.
- Se vor lua în considerare atât implementările concepute pentru compilatoare pe 16 biți, cât și cele pentru compilatoare pe 32 de biți.

**SUBIECTUL I**

**(30 de puncte)**

1.	b	4p.	
2.	a.	6p.	Se acordă numai 3p. pentru un răspuns parțial corect care să includă menționarea secvenței formată din cel puțin primii trei termeni (1, 1, 2).
	b.	6p.	Se acordă câte 3p. pentru fiecare valoare corectă.
	c.	4p.	(*) Se acordă numai 1p. dacă algoritmul are o singură structură repetitivă, principial corectă, dar nu este echivalent cu cel dat. 3p. Se va puncta orice formă corectă de structură repetitivă: <b>repetă...până când</b> , <b>repetă...cât timp</b> , <b>execută...cât timp</b> , <b>cât timp...execută</b> , <b>do...while</b> etc. (**) Două soluții posibile sunt: citește n,k pentru i←0,n-1 execută   dacă i%k=0 atunci t←t+1   ■   scrie t   ■ sau citește n,k pentru i←0,n-1 execută   scrie [i/k]+1   ■

Probă scrisă la **Informatică**

**Varianta 9**

Filiera teoretică, profilul real, specializările matematică – informatică, matematică – informatică intensiv informatică  
Filiera vocațională, profilul militar, specializarea matematică – informatică  
Barem de evaluare și de notare

<b>d. Pentru program corect</b>	<b>10p.</b>	(*) Se acordă numai 2p. dacă doar una dintre structuri este corectă.
-declarare corectă a tuturor variabilelor	1p.	
-citire corectă	1p.	
-scriere corectă	1p.	
-instrucțiune de decizie corectă	2p.	
-instrucțiuni repetitive cu test inițial corecte (*)	3p.	
-atribuiri corecte	1p.	
-corectitudine globală a programului <sup>1)</sup>	1p.	

#### SUBIECTUL al II-lea

(30 de puncte)

<b>1. d</b>	<b>4p.</b>	
<b>2. c</b>	<b>4p.</b>	
<b>3. Răspuns corect: 2 3</b>	<b>6p.</b>	Se acordă numai 3p. dacă se indică nodurile care fac parte dintr-o componentă conexă, dar nu cea care are cele mai puține noduri.
<b>4. Pentru răspuns corect (*)</b>	<b>6p.</b>	(*) O soluție posibilă este transcrierea în limbaj de programare a atribuirii
-accesare a unui element al tabloului	2p.	$A_{i,j} \leftarrow i+j$
-expresie corectă de calcul al valorii elementului	3p.	
-atribuire corectă	1p.	
<b>5. Pentru program corect</b>	<b>10p.</b>	
-declararea corectă a tuturor variabilelor (șiruri de caractere + variabile simple)	1+1p.	
-citirea corectă a tuturor datelor	2p.	
-determinarea principal corectă a unui cuvânt care începe cu un alt cuvânt	2p.	
-numărarea cuvintelor cerute	2p.	
-afișarea rezultatului	1p.	
-corectitudinea globală a programului <sup>1)</sup>	1p.	

#### SUBIECTUL al III-lea

(30 de puncte)

<b>1. c</b>	<b>4p.</b>	
<b>2. Răspuns corect:</b>	<b>6p.</b>	(*) Se acordă câte 2p. pentru fiecare triplet menționat corect și 2p. pentru enumerarea acestora în ordinea cerută. Nu se depunțează elevii dacă soluțiile nu sunt scrise între paranteze.
(violet, albastru, galben) și (violet, albastru, verde) (*)		
<b>3. Pentru subprogram corect</b>	<b>10p.</b>	(*) Pentru limbajul Pascal, parametrul de tip tablou impune definirea, în prealabil, a unui tip adecvat.
-structură antet principal corectă	1p.	
-declarare corectă a parametrilor (de tip simplu + tablou) (*)	1+1p.	
-transfer parametri conform cerinței	1p.	
-declararea tuturor variabilelor locale	1p.	
-algoritm principal corect de inserare a unei valori în tablou	2p.	
-inserare, după fiecare număr par, a valorii indicate	2p.	
-corectitudine globală a subprogramului <sup>1)</sup>	1p.	

4.	<p><b>a. Pentru răspuns corect</b>          -coerența explicării metodei (*)          -explicarea unor elemente de eficiență din punct de vedere al timpului de executare</p>	<p><b>4p.</b>          2p.            2p.</p>	<p>(*) Se acordă punctajul chiar dacă metoda aleasă nu este eficientă</p>
	<p><b>b. Pentru program corect</b>          -operații cu fișiere: declarare, pregătire în vederea scrierii, scriere în fișier          -respectarea cerinței privind primele două cifre ale numărului          -respectarea cerinței privind ultimele două cifre ale numărului          -afișarea tuturor numerelor cerute (*),(**)          -utilizarea unui algoritm eficient (***)</p>	<p><b>6p.</b>            1p.            1p.            1p.            2p.          1p.</p>	<p>(*) Se acordă punctajul chiar dacă soluția propusă nu prezintă elemente de eficiență sau afișează numerele cifră cu cifră.          (**) Se acordă numai 1p. dacă numerele nu sunt scrise în formatul cerut sau dacă fișierul conține, în plus, și alte numere.          (***) Punctajul se acordă numai pentru un algoritm în care una dintre primele două cifre este determinată în funcție de cealaltă, și una dintre ultimele două cifre este determinată în funcție de cealaltă.          O soluție posibilă de generare este transpunerea în limbaj de programare a secvenței pseudocod:  <pre> m1←s1 dacă s1&gt;9 atunci     m1←9     ■ m2←s2 dacă s2&gt;9 atunci     m2←9     ■ pentru c1=1,m1 execută     c2←s1-c1     pentru c3=0,9 execută         pentru c4=0,m2 execută             c5←s2-c4             scrie c1·10<sup>4</sup>+c2·10<sup>3</sup>+c3·10<sup>2</sup>+c4·10+c5             ■         ■     ■ ■ </pre></p>

<sup>1)</sup> Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.