

**Examenul de bacalaureat 2012**  
**Proba E. d)**  
**Proba scrisă la INFORMATICĂ**  
**Limbajul C/C++**

**MODEL**

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** **NU** aparține reuniunii de intervale  $[-3, -1] \cup [1, 3]$ . **(4p.)**
- a. `!(x)>=-3 && x<=-1) || !(x)>=1 && x<=3)`
  - b. `!(x)>=-3 || x<=-1 || x>=1 || x<=3)`
  - c. `x<-3 || x>3 || x>-1 && x<1`
  - 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 7, iar pentru **k** se citește valoarea 3. **(6p.)**
- b. Dacă pentru variabila **k** se citește valoarea 11, scrieți cea mai mică și cea mai mare dintre valorile din intervalul  $[0, 99]$  care pot fi citite pentru variabila **n** astfel încât, în ambele cazuri, în urma executării algoritmului, ultimul număr care se afișează să fie 8. **(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)  
cât timp n≥1 execută  
|dacă n>k atunci i←k  
|altfel i←n  
|  
|n←n-i  
|t←1  
|cât timp i≥1 execută  
|scrie t, ' '  
|t←t+1  
|i←i-1  
|  
|
```

**SUBIECTUL al II-lea**

**(30 de puncte)**

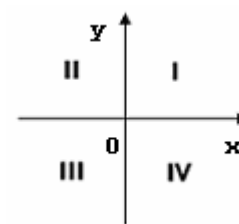
Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

- Se consideră graful neorientat cu 5 noduri, numerotate de la 1 la 5, și muchiile [1,2], [2,3], [4,5]. Pentru ca graful menționat mai sus să devină arbore este necesară: **(4p.)**
  - adăugarea unei noi muchii
  - eliminarea uneia dintre muchii
  - adăugarea a două noi muchii
  - eliminarea a două dintre muchii
- Se consideră un graf orientat cu 100 de vârfuri, fiecare dintre acestea având atât gradul interior cât și gradul exterior egale cu 99. Numărul maxim de arce care pot fi eliminate din graf astfel încât, în graful parțial obținut, între oricare două vârfuri să existe cel puțin un arc, este: **(4p.)**
  - 9801
  - 4950
  - 900
  - 50

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

- Se consideră declarațiile alăturate, în care variabila `c` memorează raza și coordonatele, în sistemul de coordonate  $xOy$ , ale centrului unui cerc.  

```
struct punct {  
    int x, y;  
};  
struct cerc {  
    struct punct centru;  
    float raza;  
} c;
```



Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă centrul cercului se află în cadranul al II-lea al sistemului de coordonate și nu pe axele acestuia. **(6p.)**

- 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	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0

**(6p.)**

- 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 se încheie cu primul cuvânt citit.

**Exemplu:** dacă `n=5` și cuvintele citite sunt:

oare  
soarele  
stropitoare  
oarecare  
cicoare

pe ecran se afișează 2 (deoarece cuvintele stropitoare și cicoare se încheie cu oare). **(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Se consideră subprogramul recursiv **CM**, definit alăturat. Pentru a returna valoarea celui mai mare divizor comun al numerelor naturale 100 și 330, apelul subprogramului poate fi: **(4p.)**
- |   |                           |
|---|---------------------------|
| <pre>int CM (int x, int y, int z) { if(x%z==0 &amp;&amp; y%z==0) return z;   else return CM(x,y,z-1); }</pre> |                           |
| a. <b>CM(1,100,330)</b>   | b. <b>CM(100,330,1)</b>   |
| c. <b>CM(100,330,100)</b>   | d. <b>CM(330,330,100)</b> |

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

2. Având la dispoziție cinci fructe diferite, un **măr**, o **pară**, o **gutui**, o **caisă**, o **piersică**, se utilizează metoda backtracking pentru a obține toate posibilitățile de a forma grupe de câte trei fructe, știind că în cadrul unei grupe nu contează ordinea de așezare a fructelor. Primele patru soluții obținute sunt, în această ordine: (**măr, pară, gutui**), (**măr, pară, caisă**), (**măr, pară, piersică**), (**măr, gutui, caisă**). 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, inserând înainte de fiecare termen par al șirului numărul obținut prin împărțirea la 2 a valorii acestuia 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  $a=(1, \underline{4}, 5, 3, \underline{82}, \underline{6}, \underline{2})$  atunci după apel  $n=11$  și  $a=(1, \underline{2}, \underline{4}, 5, 3, \underline{41}, \underline{82}, \underline{3}, \underline{6}, \underline{1}, \underline{2})$ . **(10p.)**
4. Numim număr de tip **palindrom** un număr care are aceeași valoare dacă este citit de la stânga la dreapta sau de la dreapta la stânga.
- Numim număr de tip **dublu palindrom** o valoare de tip palindrom cu număr par de cifre în care fiecare cifră de rang impar este egală cu cifra alăturată din dreapta ei.
- Exemplu:** 11111111, 22777722 sunt numere de tip dublu palindrom; 12122121 este număr de tip palindrom, dar nu și dublu palindrom.
- Se cere scrierea în fișierul **BAC.TXT**, fiecare pe câte o linie, a tuturor numerelor naturale de tip dublu palindrom, cu exact 8 cifre. Numerele de tip dublu palindrom sunt scrise în ordine strict crescătoare, iar pentru determinarea acestora se utilizează un algoritm eficient din punct de vedere al timpului de executare.
- a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
- b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**