

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

Varianta 1

*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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

**SUBIECTUL I** **(30 de puncte)**

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

1. Indicați expresia care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă  $x$  aparține mulțimii  $\{1, 2, 3\}$ . **(4p.)**

a.  $x==1 \ \&\& \ x==2 \ \&\& \ x==3$

b.  $!( \ x<=1 \ \&\& \ x>=3 \ )$

c.  $x==1 \ || \ x==2 \ || \ x==3$

d.  $!( \ x<=1 \ || \ x>=3 \ )$

2. Se consideră algoritmul alăturat, reprezentat în pseudocod.

S-a notat cu  $x \div y$  restul împărțirii numărului natural  $x$  la numărul natural nenul  $y$  și cu  $[z]$  partea întregă a numărului real  $z$ .

a) Scrieți numărul afișat dacă pentru variabila  $n$  se citește valoarea 56413. **(6p.)**

b) Scrieți toate numerele naturale, fiecare având **exact** patru cifre, care pot fi citite pentru variabila  $n$  astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea să se afișeze valoarea 40. **(4p.)**

```
citește n (număr natural)
m←0
p←1
cât timp n>0 execută
┌dacă n%2≠0 atunci
│ n←n-1
│ ■
│ m←m+(n%10)*p
│ n←[n/10]
│ p←p*10
│ ■
scrie m
```

c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă de alt tip. **(6p.)**

d) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

**SUBIECTUL al II-lea**

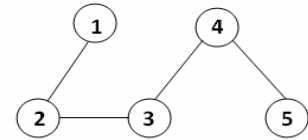
**(30 de puncte)**

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

1. Se consideră graful orientat cu 7 vârfuri, numerotate de la 1 la 7, și arcele  $(1,2)$ ,  $(2,3)$ ,  $(3,4)$ ,  $(4,1)$ ,  $(5,4)$ ,  $(7,5)$ ,  $(5,6)$ . Numărul de vârfuri ale grafului care au gradul extern egal cu gradul intern este: **(4p.)**

- a. 2                                      b. 3                                      c. 4                                      d. 5

2. Se consideră graful neorientat cu cinci noduri, reprezentat alăturat. Numărul minim de muchii ce trebuie adăugate astfel încât, în graful obținut, între oricare două noduri distincte să existe cel puțin un lanț elementar de lungime 2, este: **(4p.)**



- a. 1                                      b. 2                                      c. 3                                      d. 4

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

3. Se consideră declarațiile alăturate, în care variabila  $s$  memorează coordonatele, în sistemul de coordonate  $xOy$ , ale extremităților unui segment. Scrieți o instrucțiune care afișează pe ecran pătratul lungimii segmentului corespunzător variabilei  $s$ . **(6p.)**

```
struct punct {  
    float x,y;  
};  
struct segment {  
    punct A,B;  
}s;
```

4. În secvența de instrucțiuni de mai jos variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $A$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila  $A$  să aibă elementele din figura de mai jos.

```
for(i=1;i<=5;i++)  
    for(j=1;j<=5;j++)  
        .....  
(6p.)                                       $\left| \begin{array}{ccccc} 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \end{array} \right.$ 
```

5. Se consideră două șiruri de caractere  $a$ , de lungime  $na$  și  $b$ , de lungime  $nb$ . Șirul  $a$  este numit **prefix** al șirului  $b$  dacă  $na \leq nb$  și subșirul lui  $b$  determinat de primele sale  $na$  caractere coincide cu  $a$ . Șirul  $a$  este numit **sufix** al șirului  $b$  dacă  $na \leq nb$  și subșirul lui  $b$  determinat de ultimele sale  $na$  caractere coincide cu  $a$ .

Scrieți un program C/C++ care citește de la tastatură două cuvinte, fiecare fiind format din cel mult 30 de caractere, numai litere mici ale alfabetului englez. După primul cuvânt se tastează Enter. Programul afișează pe ecran, separate prin câte un spațiu, toate sufixele primului cuvânt care sunt prefixe ale celui de al doilea cuvânt, ca în exemplu. Dacă nu există astfel de subșiruri, se afișează pe ecran mesajul **NU EXISTA**.

**Exemple:**

dacă se citesc, în ordine, cuvintele

**rebele**

**elegant**

se afișează (nu neapărat în această ordine) subșirurile: **e ele**

iar dacă se citesc, în ordine, cuvintele

**strai**

**rai**

se afișează subșirul **rai**

**(10p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

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

1. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile de 5 cifre din mulțimea  $\{0, 1, 2\}$ , cu proprietatea că nu există două cifre pare pe poziții consecutive. Primele opt șiruri generate sunt, în această ordine: 01010, 01011, 01012, 01101, 01110, 01111, 01112, 01121. Al nouălea șir generat este: **(4p.)**
- a. 01120                      b. 01201                      c. 01210                      d. 10101

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

2. Se consideră subprogramul `f`, definit alăturat. Scrieți ce se afișează în urma apelului de mai jos. `f(9)`; **(6p.)**
- ```
void f(int x)
{ cout<<x; | printf("%d",x);
  if(x>0)
  { if(x%2==0)
    cout<<'*'; | printf("");
    f(x/2);
  }
}
```
3. Se consideră subprogramul `dublu`, cu doi parametri:
- `n`, prin care primește o valoare naturală  $2 < n < 50$ ;
  - `v`, prin care primește un tablou unidimensional cu `n` elemente, numere întregi cu cel mult 4 cifre. Cel puțin unul dintre elementele tabloului este strict pozitiv.
- După fiecare element strict pozitiv din tablou, subprogramul inserează câte un nou element, cu aceeași valoare, ca în exemplu. Tabloul modificat, precum și valoarea actualizată a lui `n`, sunt furnizate tot prin parametrii `v`, respectiv `n`. Scrieți definiția completă a subprogramului.
- Exemplu:** dacă `n=6` și `v=(4, -5, 0, 9, 9, -2)`, atunci după apel `n=9`, iar `v=(4, 4, -5, 0, 9, 9, 9, 9, -2)`. **(10p.)**
4. Un număr natural cu cel puțin două cifre se numește **x-ordonat** dacă toate cifrele sale sunt în ordine crescătoare și valoarea absolută a diferenței dintre oricare două cifre aflate pe poziții consecutive este egală cu `x`.
- Exemple:** numărul 2468 este 2-ordonat, numărul 147 este 3-ordonat; numerele 179 sau 131 nu sunt de tipul menționat.
- Se citește de la tastatură un număr natural `x` ( $1 \leq x \leq 8$ ) și se cere scrierea în fișierul `BAC.TXT` a tuturor numerelor naturale distincte **x-ordonate**. Fiecare număr este scris pe câte o linie a fișierului.
- Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul 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.)**