

# Varianta 1 Subiectul I

ex.1

1. Variabila  $x$  este de tip real. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă numărul real memorat în variabila  $x$  aparține intervalului  $(5, 8]$ ? (4p.)
- a.  $(x < 8) \&\& (x \geq 5)$
- b.  $(x <= 8) || (x > 5)$
- c.  $(x > 8) || (x <= 5)$
- d.  $(x <= 8) \&\& (x > 5)$

Raspuns: d)

ex.2

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

2. Se consideră algoritmul alăturat, descris în pseudocod.

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

- a) Scrieți valoarea care se va afișa dacă se citește  $n=103456$ . (6p.)

- b) Scrieți toate numere naturale **impare**, distincte, fiecare având **exact** două cifre, care pot fi citite pentru variabila  $n$  astfel încât să se afișeze valoarea 3. (4p.)

- 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) Scrieti programul c/c++ corespunzător algoritmului dat.

```

citește n (număr natural)
z ← 0
p ← 1
cât timp n > 0 execută
    c ← n % 10
    n ← [n / 10]
    dacă c % 3 = 0 atunci
        z ← z + p * (9 - c)
        p ← p * 10
scrie z

```

(10p.)

a)

citeste 103456

$$z = 0$$

$p = 1$

103456>0 adv

$c = 6$

n = 10345

c%3 = 0 adv

$$z = z + p^*(9 - c) = 0 + 1^*(9 - 6) = 3$$

$p = 10$

10345>0 adv

$c = 5$

n = 1034  
c%3 fals

1034>0 adv  
c = 4  
n = 103  
c%3 fals

103>0 adv  
c = 3  
n = 10  
c%3 adv  
 $z = z + p * (9 - c) = 3 + 10 * (9 - 3) = 3 + 60 = 63$

10>0 adv  
c = 0  
n = 1  
c%3 adv ( $0 \% 3 = 0$ )  
z = 963

1>0 adv  
c = 1  
n = 0  
c%3 fals

0>0 fals  
iesim din while

**Raspuns: 963**

b)

**b)** Scrieți toate numere naturale **impare**, distincte, fiecare având **exact** două cifre, care pot fi citite pentru variabila **n** astfel încât să se afișeze valoarea 3. **(4p.)**

**Raspuns: 61 65 67**

c)

- 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.)**

citeste n ( numar natural )

z <- 0

p <- 1

daca n>0 atunci

executa

{

c <- n%10

n <- [n/10]

daca c%3=0 atunci

{

z <- z + p\*(9-c)

p <- p\*10

}

}

scrie z

d)

cu o structură repetitivă de un tip.

(5p.)

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

```
45  /*
46  using namespace std;
47  int n,p,z;
48  int main()
49  {
50  cin >> n;
51  z = 0;
52  p = 1;
53  while(n>0) {
54      int c = n%10;
55      n/=10;
56      if(c%3==0)
57      {
58          z = z + p*(9-c);
59          p = p*10;
60      }
61  }
62  cout << z;
63  }
64
```

## Subiectul II

### ex.1

1. Se consideră o coadă în care inițial au fost introduse, în această ordine, elementele cu valorile 1 și 2: 

1	2
---	---

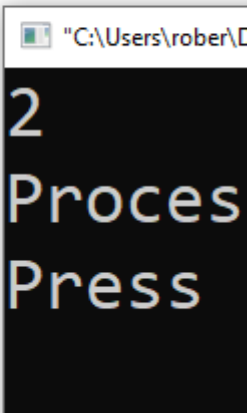
. Se notează cu **AD** (**x**) operația prin care se adaugă elementul cu valoarea **x** în coadă și cu **EL** operația prin care se elimină un element din coadă. Câte elemente va conține coada în urma executării secvenței de operații: **AD** (4) ; **EL** ; **EL** ; **AD** (5) ; **EL** ; **AD** (3) ? **(4p.)**
- a. 3                      b. 1                      c. 2                      d. 5

Raspuns:

c)

- Avem 2 elemente in coada.
- Se adauga 4. Acum avem 3 elemente in coada.
- Se sterge 1, primul element. Acum avem 2 elemente in coada.
- Se sterge 2, primul element. Acum avem 1 element in coada.
- Se adauga 5. Acum avem 2 elemente in coada.
- Se elimina 4. Acum avem 1 element in coada.
- Se adauga 3. Acum avem 2 elemente in coada.
- La sfarsit, vom avea 2 elemente in coada.

```
45 | #include <bits/stdc++.h>
46 | using namespace std;
47 | queue<int>Q;
48 | int main()
49 | {
50 |     Q.push(1);
51 |     Q.push(2);
52 |     Q.push(4);
53 |     Q.pop();
54 |     Q.pop();
55 |     Q.push(5);
56 |     Q.pop();
57 |     Q.push(3);
58 |     cout << Q.size();
59 |
60 | }
61 |
```



## ex.2

2. Câte grafuri neorientate, distincte, cu 4 vârfuri, se pot construi? Două grafuri se consideră distincte dacă matricele lor de adiacență sunt diferite. (4p.)
- a. 24      b. 4      c.  $4^6$       d.  $2^6$

Raspuns: d)

Explicatie:

- Teoria Grafurilor:

**Propoziție:** Sunt  $2^{\frac{n*(n-1)}{2}}$  grafuri neorientate distincte cu  $n$  vârfuri.

- Inlocuim  $n$  cu 4:  $2^{(4*3/2)} = 2^6$

## ex.3

3. În declarația alăturată, câmpurile  $x$  și  $y$  ale înregistrării pot memora coordonatele carteziene ale unui punct din planul  $xOy$ . Scrieți o secvență de instrucțiuni prin executarea căreia se memorează în variabila  $C$  coordonatele mijlocului segmentului cu capetele în punctele ale căror coordonate sunt memorate în variabilele  $A$  și  $B$ . (6p.)

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

Raspuns:

$C.x = (A.x+B.x)/2;$

$C.y = (A.y+B.y)/2;$

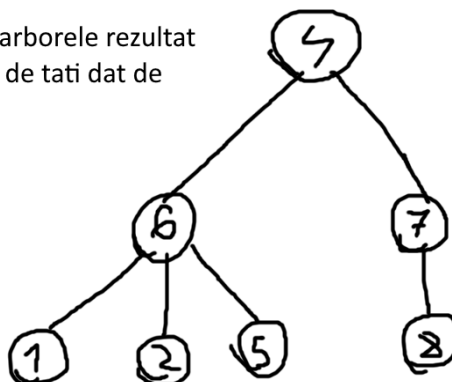
## ex.4

4. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț format din noduri distincte care are una dintre extremități în rădăcina arborelui. Scrieți care este înălțimea și care sunt frunzele arborelui descris prin următorul vector "de tați": (6, 6, 5, 0, 6, 4, 4, 7). (6p.)

Raspuns:

Varianta 1, Subiectul II, exercitiul 3.

Acesta este arborele rezultat din vectorul de tați dat de exercitiu.



Înălțimea arborelui este 3 (Cele mai lungi lanturi [4,7,8]/[4,6,1/2/5] au lungimea 3)  
Numarul de frunze este 4 (1,2,5,8)

## ex.5

5. Scrieți un program C/C++ care citește de la tastatură două numere naturale nenule  $n$  și  $m$  ( $2 \leq m \leq 10$ ,  $2 \leq n \leq 10$ ) și care construiește în memorie și apoi afișează o matrice  $A$  cu  $n$  linii (numerotate de la 1 la  $n$ ) și  $m$  coloane (numerotate de la 1 la  $m$ ) cu proprietatea că fiecare element  $A_{ij}$  memorează cea mai mică dintre valorile indicilor  $i$  și  $j$  ( $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ). Matricea se va afișa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru  $n=4$  și  $m=5$  se va afișa matricea alăturată.

(10p.)

1	1	1	1	1
1	2	2	2	2
1	2	3	3	3
1	2	3	4	4

```

45 #include <bits/stdc++.h>
46 using namespace std;
47 int mat[11][11];
48 int n,m;
49 int main()
50 {
51     cin >> n >> m;
52     for(int i=1;i<=n;i++)
53         for(int j=1;j<=m;j++)
54             mat[i][j] = min(i,j);
55
56
57     for(int i=1;i<=n;i++){
58         for(int j=1;j<=m;j++)
59             cout << mat[i][j] << " ";
60         cout << "\n";
61     }
62 }
63

```

P.S: Posibilitatea unei depunctari la bacalaureat pentru folosirea functiei min este o realitate! Astfel, aceasta se poate inlocui cu:

```

if(i<j)
    mat[i][j] = i;
else mat[i][j] = j;

```

## Subiectul III

ex.1

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

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A = \{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab, abac, abad, abba, abbb, abbc, abbd, abbe**. Câte dintre cuvintele generate încep cu litera **b** și se termină cu litera **e**? **(4p.)**
- a. 9                      b. 15                      c. 12                      d. 20

Raspuns: b)

- Urasc faptul ca problema precizeaza "utilizand metoda backtracking". Acest lucru nu afecteaza sub nicio forma rezultatul final. In realitate, este un bun creator de confuzie in mintea unui elev stresat in timpul examenului.
- Cuvantul obtinut va fi de forma  $b x y e$
- $x$  poate fi inlocuit cu toate cinci litere :  $a, b, c, d, e$ .
- $y$  poate fi inlocuit cu doar 3 litere, deoarece inlocuire acestuia cu o vocala ar invalida restrictia impusa asupra noastra (cuvantul generat nu trebuie sa contina doua vocale alaturate)
- Astfel, avem 5 posibilitati pentru o pozitie si inca 3 pentru cealalta pozitie. Rezulta un numar total de  $5 \cdot 3$  posibilitati, adica 15 cuvinte generate.



## ex.2

2. Pentru definiția de mai jos a subprogramului  $f$ , ce se afișează ca urmare a apelului  $f(121,1)$ ? (6p.)

```
//c
void f(long n, int i)
{ if(n!=0)
  if(n%3>0)
    { printf("%d",i); f(n/3,i+1); }
}
```

```
//c++
void f(long n, int i)
{ if(n!=0)
  if(n%3>0)
    { cout<<i; f(n/3,i+1); }
}
```

$f(121,1)$

$n\%3 > 0$  adv

afiseaza 1

$f(40,2)$

$n\%3 > 0$  adv

afiseaza 2

$f(13,3)$

$n\%3 > 0$  adv

afiseaza 3

$f(4,4)$

$n\%3 > 0$  adv

afiseaza 4

$f(1,5)$

$n\%3 > 0$  adv

afiseaza 5

$f(0,6)$

$n \neq 0$  fals

se iese din subprogram

**Se afiseaza 1 2 3 4 5**

### ex.3

3. Fișierul text `bac.txt` conține, pe o singură linie, cel mult 1000 de numere naturale nenule cu cel mult 4 cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural nenul  $n$  ( $n \leq 999$ ) și numerele din fișierul `bac.txt` și care afișează pe ecran, separate prin câte un spațiu, toate numerele din fișier care sunt divizibile cu  $n$ . Dacă fișierul nu conține niciun astfel de număr, atunci se va afișa pe ecran mesajul `NU EXISTA`.

**Exemplu:** dacă fișierul `bac.txt` conține numerele: 3 100 40 70 25 5 80 6 3798, pentru  $n=10$  atunci pe ecran se va afișa: 100 40 70 80 (10p.)

```
81  #include <bits/stdc++.h>
82  using namespace std;
83  ifstream fin("bac.txt");
84  int n,x;
85  int main()
86  {
87      fin >> n;
88      bool ok = false;
89      while(fin >> x){
90          if(x%n==0){
91              cout << x << " ";
92              ok = true;}
93      }
94      if(ok==false)
95          cout << "NU EXISTA";
96      }
97  }
```

### ex.4

4. Subprogramul `sub`, cu trei parametri, primește prin intermediul parametrilor:
- $v$  un tablou unidimensional cu cel mult 100 de componente ce memorează numere întregi de cel mult 4 cifre fiecare;
  - $n$  un număr natural nenul mai mic sau egal cu 100 ce reprezintă numărul efectiv de componente ale tabloului primit prin intermediul parametrului  $v$ ;
  - $a$  un număr întreg cu cel mult 4 cifre.

Subprogramul `sub` returnează numărul componentelor tabloului primit prin intermediul parametrului  $v$  ale căror valori sunt strict mai mici decât valoarea parametrului  $a$ .

**Exemplu:** pentru valorile  $n=5$ ,  $v=(1,21,9,21,403)$ ,  $a=20$  ale parametrilor, în urma apelului, subprogramului `sub` va returna valoarea 2.

a) Scrieți definiția completă a subprogramului `sub`. (4p.)

b) Să se scrie un program C/C++ care să citească de la tastatură un număr natural nenul  $n$  ( $n \leq 100$ ) și  $n$  numere întregi, fiecare având cel mult 4 cifre, și care, folosind apeluri utile ale subprogramului `sub`, să afișeze pe ecran mesajul `DA` dacă oricare două dintre cele  $n$  numere întregi citite sunt distincte două câte două, sau mesajul `NU` în caz contrar.

**Exemplu:** pentru  $n=6$  și cele  $n$  numere citite de la tastatură: 47 183 69 8 134 -56 se va afișa pe ecran mesajul `DA` (6p.)

a)

```
85 int sub(int v[101],int n, int a){
86     int s=0;
87     for(int i=1;i<=n;i++)
88         if(v[i]<a)
89             s++;
90     return s;
91 }
```

b)

```
81 #include <bits/stdc++.h>
82 using namespace std;
83 ifstream fin("bac.txt");
84
85 int sub(int v[101],int n, int a){
86     int s=0;
87     for(int i=1;i<=n;i++)
88         if(v[i]<a)
89             s++;
90     return s;
91 }
92
93 int n;
94 int v[101];
95 int main()
96 {
97     cin >> n;
98     for(int i=1;i<=n;i++)
99         cin >> v[i];
100
101     //Sortam elementele crescator.
102     for(int i=1;i<=n;i++)
103         for(int j=i+1;j<=n;j++)
104             if(v[j]<v[i]) swap(v[i],v[j]);
105
106     //Apelam functia sub pentru a afla numarul de element mai mici ca v[i]. Datorita faptului ca elementele sunt ordonate
107     //crescator, in cazul in care nu exista elemente egale intre ele,
108     //subprogramul sub va trebui sa returneze, pentru verificarea fiecarui element de la 1 la n, in ordine crescatoare,
109     //numerele 0,1,2,...,n-1, si se afiseaza DA. Altfel, se afiseaza NU.
110     for(int i=1;i<=n;i++){
111         if(sub(v,n,v[i])!=i-1)
112             {cout << "NU"; return 0;}
113     }
114     cout << "DA";
115 }
```