

# Varianta 36

## Subiectul I

ex.1

### Subiectul I (30 de puncte)

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

- |  |   |
|--|---|
| <p>1. În secvența de instrucțiuni alăturată, variabilele <math>x</math> și <math>y</math> sunt de tip <code>int</code>. Care este valoarea pe care trebuie să o aibă inițial variabila <math>x</math> dacă la finalul executării secvenței s-a afișat un singur caracter asterisc (*)? (4p.)</p> <p>a. 0                                      b. 2                                      c. 1                                      d. 4</p> | <pre>y=x; while (x&lt;=3) {     cout&lt;&lt;"*";   printf("*");     y=y+1; x=x+y; }</pre> |
|--|---|

Raspuns corect: b) 2

ex.2

- |   |  |
|---|--|
| <p>2. Se consideră algoritmul alăturat descris în pseudocod.</p> <p>S-a notat cu <math>x \% y</math> restul împărțirii numărului natural <math>x</math> la numărul natural nenul <math>y</math> și cu <math>[z]</math> partea întreagă a numărului real <math>z</math>.</p> <p>a) Scrieți ce valoare se va afișa dacă se citesc, în ordine, următoarele valori : 114, 123, 517, 3312, 14, 412, 22, 0. (6p.)</p> <p>b) Scrieți ce valoare se va afișa dacă se citesc, în ordine, primele 99 de numere naturale nenule, urmate de 0 (adică 1, 2, 3, 4, ..., 98, 99, 0). (4p.)</p> <p>c) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura <code>cât timp...execută</code> cu o structură repetitivă de un alt tip. (6p.)</p> <p>d) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)</p> | <pre>s ← 0 citește v (valoare naturală) cât timp v ≠ 0 execută     a ← v % 10     b ← [v/10] % 10     s ← s + a * 10 + b     citește v scrie s</pre> |
|---|--|

a)

Raspuns: **249**

Explicatie:

Algoritmul prezentat face cateva operatii dupa citirea fiecarui numar:

1. Inmulteste ultima cifra a numarului citit ( $v$ ) cu 10
2. Adauga la numarul obtinut cifra zecilor numarului citit ( $v$ )

3. Numarul final se adauga la s

Astfel:

$$114 \rightarrow s = 41$$

$$123 \rightarrow s = 41 + 32$$

$$517 \rightarrow s = 41 + 32 + 71$$

...

s.s.m.d. ajungem la **249**

b)

- b) Scrieți ce valoare se va afișa dacă se citesc, în ordine, primele 99 de numere naturale nenule, urmate de 0 (adică 1, 2, 3, 4, ..., 98, 99, 0). (4p.)

Raspuns:

Se va afisa  $1+2+3+\dots+98+99$ , adica  $99 \cdot 100 / 2 = \mathbf{4950}$

Prin introducerea tuturor numerelor  $>9$ , cu exceptia celor divizibile cu 10, rezultatul va fi numarul propriu-zis. In cazul in care numarul introdus este divizibil cu 10, rezultatul va fi cifra zecilor numarului specificat anterior, iar in cazul cifrelor (1,2,3,...,8,9), rezultatul afisat va fi cifra inmultita cu 10. Astfel, dupa introducerea tuturor numerelor de la 1 la 99, se va afisa  $1+2+3+\dots+98+99$  (**=4950**)

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 un alt tip. (6p.)

s<-0

citeste v

daca v!=0 atunci

|executa

||a<-v%10

||b<-[v/10]%10

||s<-s+a\*10+b

||citeste v

|-cat timp v!=0

scrie s

d)

- d) Scrieți programul c/c++ corespunzător algoritmului dat.

```
main() : int
//Varianta 36, Subiectul I, ex.2 d)
#include <bits/stdc++.h>
using namespace std;
int a,b,s;
int main()
{
    int s = 0;
    unsigned int v;
    cin >> v;
    while(v) {
        a = v%10;
        b = (v/10)%10;
        s = s + a*10 + b;
        cin >> v;
    }
    cout << s;
}
```

## Subiectul II

ex.1

- |   |  |
|---|--|
| <p>1. Știind că în urma executării secvenței alăturate s-a afișat succesiunea de caractere <b>EXAMEN</b>, care este șirul de caractere memorat de variabila <b>s</b>? (6p.)</p> <p>a. <b>ENXAME</b>                      b. <b>EAENMX</b>                      c. <b>NEEXMA</b>                      d. <b>NEMAXE</b></p> | <pre>x=strlen(s);<br/>for (i=0;i&lt;x/2;i++)<br/>    cout&lt;&lt;s[i]&lt;&lt;s[x-i-1];<br/>      printf("%c%c",s[i],s[x-i-1]);</pre> |
|---|--|

Raspuns: b)

ex.2

2. Într-o listă liniară simplu înlănțuită, alocată dinamic, fiecare element reține în câmpul **adr** adresa următorului element din listă sau **NULL** dacă nu există un element următor, iar în câmpul **info** un număr întreg. Adresa primului element al listei este memorată în variabila **p**. Știind că lista conține exact 4 elemente, atunci expresia **p->adr->info** reprezintă: (4p.)
- a. adresa celui de al doilea element                      b. adresa celui de al treilea element  
c. valoarea memorată în al doilea element                      d. valoarea memorată în al treilea element

Raspuns: c)

ex.3

3. Se consideră un graf neorientat cu 7 noduri numerotate de la 1 la 7 și muchiile **[1,2]**, **[1,3]**, **[2,3]**, **[2,4]**, **[2,5]**, **[2,6]**, **[4,6]**, **[5,7]**, **[6,7]**. Care este numărul minim de muchii ce trebuie adăugate astfel încât graful să devină eulerian și care sunt aceste muchii? (6p.)

Nr. minim de muchii adăugate este 3

Acestea pot fi [2,6],[6,1],[1,7]

Lantul eulerian aparut este [5,2,1,3,2,4,6,2,7,6,1,7,5]

ex.4

4. Câte muchii trebuie eliminate dintr-un graf neorientat complet cu 20 de noduri, pentru ca graful parțial obținut să fie arbore? (4p.)

Raspuns:

171

Un graf neorientat complet cu 20 de noduri are 190 de muchii

Un arbore cu 20 de noduri are 19 muchii

$190 - 19 = 171$

ex.5

5. Se consideră o matrice cu  $n$  linii și  $m$  coloane ( $1 \leq n \leq 30$ ,  $1 \leq m \leq 30$ ), ce memorează numere întregi de cel mult 4 cifre fiecare. Scrieți un program C/C++ care citește de la tastatură valorile  $n$ ,  $m$  și elementele matricei și care afișează pe ecran, separate prin câte un spațiu, valorile minime de pe fiecare coloană, în ordine de la ultima la prima coloană.

**Exemplu:** pentru  $n=4$ ,  $m=4$  și matricea alăturată se vor afișa pe ecran valorile 3 7 2 3. (10p.)

3	4	90	10
25	2	7	9
18	3	10	4
3	7	20	3

```
1 //Varianta 36, Subiectul II, ex.5
2 #include <bits/stdc++.h>
3 using namespace std;
4 int mat[31][31];
5 int main()
6 {
7     int n,m;
8     cin >> n >> m;
9     for(int i=1;i<=n;i++)
10         for(int j=1;j<=m;j++)
11             cin >> mat[i][j];
12     for(int i=m;i>=1;i--){
13         int minim_coloana = 10000;
14         for(int j=n;j>=1;j--){
15             if(mat[j][i]<minim_coloana)
16                 minim_coloana=mat[j][i];
17         }
18         cout << minim_coloana << " ";
19     }
20 }
21
```

## Subiectul III

ex.1

1. Subprogramul `scif` returnează suma cifrelor unui număr natural transmis ca parametru. Care este valoarea expresiei `scif(scif(518)+scif(518))` ? (4p.)
- a. 14                      b. 10                      c. 28                      d. 1

Raspuns: b)

ex.2

2. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 9 ca sumă a cel puțin două numere naturale nenule distincte. Termenii fiecărei sume sunt în ordine strict crescătoare. Soluțiile se generează în ordinea: 1+2+6, 1+3+5, 1+8, 2+3+4, 2+7, 3+6 și 4+5. Se aplică exact aceeași metodă pentru scrierea lui 12. Scrieți, în ordinea generării, toate soluțiile de forma 2+... (6p.)

Raspuns:

**2+3+7**

**2+4+6**

**2+10**

### ex.3

Subprogramul `cifra` primește prin intermediul parametrului `a` un număr natural cu cel mult 4 cifre și returnează ultima cifră pară a sa. Dacă numărul nu conține cifre pare, subprogramul returnează valoarea -1. De exemplu, dacă `a=8345`, subprogramul va returna 4.

a) Să se scrie definiția completă a subprogramului `cifra`. (10p.)

b) Pe prima linie a fișierului `bac.in` se află un număr natural nenul `n` ( $n \leq 15000$ ), iar pe a doua linie a fișierului se află un șir de `n` numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre.

Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran, folosind apeluri utile ale subprogramului `cifra`, cel mai mare număr care se poate forma cu ultimele cifre pare ale fiecărui element, dacă acestea există. Alegeți o metodă de rezolvare eficientă ca timp de executare. Dacă toate numerele de pe a doua linie a fișierului au numai cifre impare, programul va afișa mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul `bac.in` are conținutul

7	
369 113 2 0 33 1354 42	

alăturat, pe ecran se va afișa: **64220** (6p.)

c) Descrieți succint în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 - 4 rânduri). (4p.)

a)

```

6  unsigned int cifra(unsigned int a) {
7      if(a==0)
8          return 0;
9      while(a) {
10         if((a%10)%2==0)
11             return a%10;
12         a/=10;
13     }

```

(unsigned int - variabila de tip numar natural)

b)

```

1  //Varianta 36, Subiectul III, ex.3 b
2  #include <bits/stdc++.h>
3  using namespace std;
4  ifstream fin("bac.in");
5  int v[10001];
6  unsigned int cifra(int a) {
7      if(a==0)
8          return 0;
9      while(a) {
10         if((a%10)%2==0)
11             return a%10;
12         a/=10;
13     }
14     return -1;
15 }
16 int n;
17 int main()
18 {
19     int x=-1;
20     fin >> n;

```

```

21     for(int i=1;i<=n;i++){
22         fin >> x;
23         if(cifra(x)!=-1)
24             v[cifra(x)]++;
25     }
26     bool ok = false;
27     bool only0 = true;
28     for(int i=9;i>=0;i--){
29         if(v[i]>0){
30             ok=true;
31             if(i>0)
32                 only0 = false;
33         }
34         while(v[i]){
35             if(only0==true)
36                 {cout << 0; return 0;}
37             v[i]--;
38             cout << i;
39         }
40     }
41     if(ok==false)
42         cout << "NU EXISTA";
43 }

```

c)

Am rezolvat aceasta problema folosind un **vector de frecventa**. Fiecare numar se citește, după care **vectorul cu indicele rezultatului funcției cifra(x) este incrementat cu 1**.

Este o rezolvare **eficientă** deoarece **se pastrează doar ultima cifră pentru a fiecărui număr**, nu și numărul propriu-zis. Astfel, rezultatul final se poate afla rapid, deoarece verificăm valoarea fiecărui element al vectorului de frecvență, de la cel mai mare la cel mai mic și formăm rezultatul. (evident cu verificarea cazului în care avem numai 0-uri)