

MODEL TEST 7

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări 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). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila x memorează un număr natural de exact 5 cifre. Care dintre următoarele expresii are ca valoare un număr natural format din cele 3 cifre situate în mijlocul numărului memorat în variabila x ?

- a. $x\%10$ b. $x/10$
c. $x/10\%1000$ d. $x\%1000/10$

2. Aplicând algoritmul de căutare binară pentru căutarea unei valori x în tabloul unidimensional (2,4,5,6,8,12,16), se fac exact trei comparații. Care dintre următoarele valori poate fi valoarea variabilei x ?

- a. 2,5 b. 4,12 c. 8,16 d. 2,5,8,16

3. Ce va afișa pe ecran următorul program?

```
#include <iostream>
using namespace std;
int main()
{   int n=10, i;
    int a[]={11,2,3,44,555,66,7,8,9,1};
    for (i=n/2; i; i--) a[i+n/2-1]=a[i];
    for (i=0; i<n; i++) cout<<a[i];
    return 0;
}
```

- a. 112344555667891 c. 234455566112344555
b. 112344555234455566 d. 445552344555661123

4. Ce va afișa pe ecran următorul program, dacă se citesc de la tastatură valorile {12,1,2,2,3,1,2,1,3,4,1,2,1}?

```
int v[20],n,i,j,k=0;
cout<<"n=";cin>>n;
    for(i=0;i<n;i++)
cin>>v[i];
    for(i=0;i<n-1;i++)
    for(j=i+1;j<n;j++)
    if(v[i]==v[j])
        k++;
cout<<k;
```

- a. 17 b. 20 c. 4 d. 3

5. Indicați ce valoare are expresia C++ următoare:

$\text{pow}(x,2) - \text{sqrt}(x+9) / \text{abs}(x+3)$, dacă variabila x are valoarea -5 .

- a. 11 b. 11.5 c. 24 d. 0

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

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

a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește numărul **29357**. (6p.)

b. Scrieți o valoare de **5** cifre care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze **123**. (6p.)

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

d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină alt tip de structuri repetitive. (6p.)

citește n (număr natural)

$x \leftarrow 0$

repetă

$x \leftarrow x * 10 + n \% 10$

$n \leftarrow [n/10]$

până când $n = 0$

repetă

$n \leftarrow n * 10 + x \% 10$

$x \leftarrow [x/100]$

până când $x = 0$

scrie n

2. Două puncte **A** și **B** din planul xOy sunt date prin coordonatele lor carteziane X_A, Y_A , respectiv X_B, Y_B . Scrieți o expresie C++ care are valoarea **1** dacă și numai dacă punctele **A** și **B** se află pe prima bisectoare a sistemului de coordonate. (6p.)

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg. Scrieți secvența alăturată, înlocuind punctele de suspensie astfel încât, în urma execuției secvenței obținute, să se afișeze pe ecran, în această ordine, numerele de mai jos.

2 4 6 8

4 6 8

6 8

8

for($i=1; i \leq 4; i++$)

{

for(.....) cout << ' ';

for(.....)

cout << $2*j$ << ' ';

cout << endl;

}

SUBIECTUL al III-lea**(30 de puncte)**

1. Se citesc de la tastatură 3 numere:

- **n**, un număr de cel mult 6 cifre scris în baza **b1**
- **b1**, baza numărului original **n**
- **b2**, baza în care va fi trecut numărul **n**

Scrieți un algoritm în pseudocod care să treacă numărul original **n**, din baza **b1** în baza **b2**.

Exemplu: dacă **n=2020**, **b1=3** și **b2=4**, în baza 10, **n** va fi **60**, iar valoarea determinată va fi **330**. **(10p.)**

2. Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul $[1, 10^9]$, în această ordine: **n**, apoi cele **n** elemente ale unui tablou unidimensional. Numărul **n** aparține intervalului $[1, 10^3]$. Programul modifică ordinea valorilor din tablou, astfel încât valorile impare să fie ordonate crescător, valorile pare să fie ordonate descrescător și fiecare număr de tip par/impar își va păstra indicii originali.

Exemplu: $n=8$, și vectorul: **2, 0, 9, 1, 2, 0, 2, 0**

Vom afișa: **2 2 1 9 2 0 0 0**

Se observă ca numerele impare si-au păstrat indicii originali, la fel si cele pare. **(10p.)**

3. Fișierul **cifre.in** conține pe prima linie cel mult 10^6 cifre. Să se citească cifrele din fișier și să se determine prima secvență cu valori identice de lungime maximă. Secvența va fi identificată prin poziția de început, poziția de final și valoarea cifrei din secvență. Afișarea se va face pe ecran.

Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 2 0 0 1 2 2	2 0 1 9 2 0 2 0
Pe ecran	4 5 0	1 1 2

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat. **(8p.)**

REZOLVARE SUBIECTE

TEST 7

SUBIECTUL I

(20 de puncte)

1c	2d	3b	4a	5c	5x4p
----	----	----	----	----	------

SUBIECTUL al II-lea

(40 de puncte)

1a. 237

b. orice număr de 5 cifre care are **prima** cifră 1, **a treia** cifră 2 și **a cincea** cifră 3.
(de exemplu 10203)

c.

```
#include <iostream>
using namespace std;
int main()
{ int n,x;
  cin>>n;
  x=0;
  do
  { x=x*10+n%10;
    n=n/10;
  } while (n!=0);
  do
  { n=n*10+x%10;
    x=x/100;
  } while (x!=0);

  cout<<n;
  return 0;
}
```

d.

```
citește n (număr natural)
x←0
┌cât timp n≠0 execută
│ x← x*10+n%10
│ n←[n/10]
└─┘
┌cât timp x≠0 execută
│ n← n*10+n%10
│ x←[x/100]
└─┘
scrie n
```

2.XA==YA && XB==YB

3.

```
for (i=1;i<=4;i++)
{
  for (j=1;j<i;j++) cout<<' ';
  for (j=i;j<=4;j++)
    cout<<2*j<<' ';
  cout<<endl;
}
```

SUBIECTUL al III - lea**(30 de puncte)**

1.

```

Citește n,b1,b2
//formarea numarului din baza b1 in baza 10
p←1;
x←0;
Cat timp (n>0) executa
|   x←x+(n%10)*p //fiecare cifra se inmulteste cu b1^0, b1^1...
|   p←p*b1
|   n←[n/10]
|_■
//numarul x, il impartim la b2
//cu resturile sale formam numar cu fixare de cifre in fata lui
m←0
p←1
cat timp (x>0) executa
|   m←m+(x%b2)*p
|   p←p*10
|   x←[x/b2]
|_■
scrie m;

```

2.

```

#include<fstream>
#include<iostream>
using namespace std;
int n,a[1002],i,j;
int main()
{
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>a[i];
    ///metoda selecției directe cu testare de parităi
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            if (a[i]%2!=0 and a[j]%2!=0)//ambele impare-sa fie crescătoare
            {
                if (a[i]>a[j])
                    swap(a[i],a[j]);
            }
            else
            if (a[i]%2==0 and a[j]%2==0)//ambele pare-sa fie descrescătoare
            {
                if (a[i]<a[j])
                    swap(a[i],a[j]);
            }
    for(i=1;i<=n;i++)
        cout<<a[i]<<' ';
}

```

3.

a. Se citesc numerele din fișier și în timpul citirii se compară numărul curent y cu cel anterior x . În caz de egalitate se va crește lungimea, iar în caz negativ înseamnă ca am terminat o secvență. Fiecare secvență terminată în timpul citirii dar și la final de citire va fi testată pe baza lungimii ei, comparând-o cu variabila $lmax$, actualizând pe parcurs, poziția de început și final alături de valoarea din secvența maximă. Algoritmul este eficient ca timp de executare întrucât este liniar și este eficient ca spațiu de memorare întrucât nu folosește tablouri.

b.

```

#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("cifre.in");
int x,l,poz,xmax,lmax,st,dr,y;
int main()
{
    fin>>x;
    //prima secventa cu 1 element
    l=1;
    poz=1;
    lmax=0;
    while(fin>>y)
    {
        poz++;
        if (x==y) l++;
        else
        {
            //s-a incheiat o secventa
            if (l>lmax)
            {
                lmax=l;
                xmax=x;
                st=poz-l;//poz minus l(variabila)
                dr=poz-1;//poz minus 1(unu, constanta)
            }
            //pregatim noua secv
            l=1;
            x=y;
        }
    }
    //ultima secv se testeaza
    if (l>lmax)
    {
        lmax=l;
        xmax=x;
        st=poz-l+1;
        dr=poz;
    }
    cout<<st<<' '<<dr<<' '<<xmax;
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 7**

- 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.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(30 de puncte)**

Răspuns	Punctaj
1c 2c 3b 4a 5d	5x4p.

SUBIECTUL al III-lea**(40 de puncte)**

1.	a) Răspuns corect: 237	6p.	
	b) Răspuns corect: orice număr de 5 cifre care are prima cifră 1, a treia cifră 2 și a cincea cifră 3. (de exemplu 10203)	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se acordă numai 4p. dacă se folosesc tot structuri repetitive cu test final.
2.	Pentru rezolvare corectă - acces la câmpurile de pe primul nivel al înregistrării - acces la câmpurile de pe al doilea nivel al înregistrării - afișare conform condiției impuse(*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă - acces la elementele fiecărei linii - atribuire a valorilor indicate elementelor conform valorilor afișate (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*)Se acordă câte 1p. pentru atribuirea valorilor conform cerinței, pentru elementele situate pe fiecare linie.

1) Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL al III-lea**(30 de puncte)**

1	Pentru algoritm corect - citirea datelor - determinarea numerelor cerute (*) - scrierea datelor - scriere principial corectă a structurilor de control (**) 	10p. 1p. 6p. 1p. 2p	(*) Se acordă câte 3p. pentru: - Transformarea din baza b_1 în baza 10 (1pct -obținerea cifrelor, 2 pct-formarea nr) - Transformarea din baza 10 in baza b_2 (1pct -obținerea cifrelor, 2 pct-formarea nr) (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
2	Pentru program corect - declarare a unei variabile care să memoreze un tablou - citire a datelor - memorare a numerelor conform cerinței(*) - afisarea tabloului - declarare a variabilelor simple, corectitudine globală a programului1)	10p. 1p. 1p. 6p. 1p. 1p	(*) Se acordă cele 6pct. astfel: - 3p prezentarea metodei de sortare (sel. Directa) - 1p verificare valori impare - 1p verificare valori pare - 1p realizarea interschimbării
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului1)	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. Contorizam lungimea secvenței dacă numărul citit coincide cu anteriorul, la terminarea unei secvențe sau a întregii citiri, comparăm lungimea secvenței cu lungimea maximă. Nu se vor folosi vectori, citire și prelucrare în același timp. Complexitate liniară.