

5. Un graf neorientat are **100** de vârfuri și **7** componente conexe. Numărul maxim de muchii pe care le poate avea acest graf este:
- a. 3822 b. 4950 c. 8742 d. 4371

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.)**

e.
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. În declarația alăturată, variabila x memorează, pentru fiecare dintre cei 30 de elevi dintr-o clasă, numărul matricol, precum și data nașterii elevului. Scrieți o secvență de instrucțiuni C/C++ care citește de la tastatură informațiile despre primul elev memorat și afișează pe ecran numărul matricol al elevului, dacă anul nașterii este mai mic decât 2013, sau mesajul **imposibil** în caz contrar. **(6p.)**

```
struct elev
{ int nr_mat;
  struct
  { int zi,luna,an;
  }data;
}x[30];
```

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional de numere întregi cu **7** linii și **7** coloane, numerotate de la **1** la **7**, având inițial toate elementele nule. Fără a utiliza alte variabile decât cele menționate, scrieți una sau mai multe instrucțiuni care pot înlocui puntele de suspensie, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat. **(6p.)**

```
for(i=1;i<=7;i++)
  for(j=1;j<=7;j++)
    .....
1 1 1 1 1 1 2
1 1 1 1 1 2 3
1 1 1 1 2 3 3
1 1 1 2 3 3 3
1 1 2 3 3 3 3
1 2 3 3 3 3 3
2 3 3 3 3 3 3
```

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

1. Subprogramul **conversie** are trei parametri:

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

Subprogramul returnează numărul obținut în baza **b2**, acest număr va avea maxim 18 cifre.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=2020**, **b1=3** și **b2=4**, în baza 10, **n** va fi **60**, iar valoarea furnizată de subprogram este **330**.

(10 p.)

2. Un text are cel mult 200 de caractere, iar cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații. Scrieți un program Pascal/C/C++ care citește de la tastatură textul și care codifică acest text prin inversarea cuvintelor care au număr egal de vocale și consoane. Textul final va avea aceeași lungime cu textul inițial, spațiile libere dintre cuvinte se păstrează.

Exemplu: textul: **azi avem simulare la info**

Se va afișa textul: **azi meva eralumis al ofni**

(10 p.)

3. Fișierul **cifre.in** conține pe prima linie cel mult **10⁶** 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.

(2 p.)

b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat.

(8 p.)

REZOLVARE SUBIECTE
TEST 6
SUBIECTUL I

(20 de puncte)

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

SUBIECTUL al II – lea

(40 de puncte)

1. a. 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. `cin>>x[0].nr_mat>>x[0].data.zi>>x[0].data.luna>>x[0].data.an;`
`if (x[0].data.an<2013)`
`cout<<x[0].nr_mat;`
`else cout<<"imposibil";`
3. `if (i+j==8) a[i][j]=2;`
`else if (i+j<8) a[i][j]=1;`
`else a[i][j]=3;`

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

1.

long long conversie(int n, int b1, int b2)

```

{
    //formarea numarului din baza b1 in baza 10
    long long x=0,p=1,m;
    while(n)
    {
        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;
    while(x)
    {
        m=m+(x%b2)*p;
        p=p*10;
        x=x/b2;
    }
    return m;
}

```

2.

```

#include<iostream>
#include<cstring>
using namespace std;
char s[202];
int i,p1,p2,l,j,ok;
int main()
{
    cin.get(s,201,'\n');
    l=strlen(s);
    for(i=0;i<l;i++)
    {
        //delimitam cuvintele-inceput
        if ((i==0 or s[i-1]==' ') and s[i]!=' ') p1=i;
        //finalul de cuvnt
        if ((i==l-1 or s[i+1]==' ') and s[i]!=' ')
        {
            p2=i;
            //parcurgem intervalul [p1,p2] si calculam vocalele/consoanele
            ok=0;

```

```

for(j=p1;j<=p2;j++)
    if (strchr("aeiou",s[j])!=NULL) ok++;
if (ok*2==p2-p1+1)
    {
        ///dublul numarului de vocale este lungimea cuvântului, facem inversarea
        while(p1<p2)
        {
            ///interchimbare literele de la capete, spre mijlocul cuvântului
            swap(s[p1],s[p2]);
            p1++;
            p2--;
        }
    }
}
cout<<s;
}

```

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 că 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 6**

- 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**(20 de puncte)**

Răspuns	Punctaj
1c 2d 3b 4c 5d	5x4p.

SUBIECTUL al II – 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 un element al tabloului - atribuire a valorilor indicate elementelor tabloului (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*)Se acordă câte 1p. pentru atribuirea valorilor conform cerinței elementelor situate sub diagonala secundară, respectiv celor situate deasupra diagonalei secundare și 2p. pentru atribuirea valorilor conform cerinței elementelor situate pe diagonala secundară.

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

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

1	Pentru subprogram corect - antet subprogram (*) - determinarea numerelor cerute (**) - declarare a tuturor variabilelor locale, corectitudine globală a subprogramului1)	10p. 3p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (tip, nume, parametrii de intrare) conform cerinței. (**) Se acordă câte 3p. pentru: - Transformarea din baza b1 în baza 10 (1pct -obținerea cifrelor, 2 pct-formarea nr) - Transformarea din baza 10 în baza b2 (1pct -obținerea cifrelor, 2 pct-formarea nr)
2	Pentru program corect - declarare a unei variabile care să memoreze un șir de caractere - citirea textului - modificarea textului conform cerinței (*) - afișarea textului - declarare a variabilelor simple, - corectitudine globală a programului1)	10p. 1p. 1p. 5p. 1p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect - obținerea unui cuvânt - determinarea lungimii unui cuvânt, - determinarea nr de vocale - determinarea nr de consoane - inversarea unui cuvânt care respecta cerința
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este
	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)	8p. 1p. 5p. 1p. 1p.	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.