ALGORITMI CARE LUCREAZA CU TABLOURI BIDIMENSIONALE

RECAP

- Matrice
- Matrice pătratică
- Diagonală principală
- Diagonală secundară
- Vecinii unui element

Let's make sure we're talking about the same things - click for a short quiz.

1. Cupa Mondială 2022

		1	0	1
	-1		-1	0
	0	1		-1
	-1	0	1	

Se consideră grupa E de la Campionatul Mondial de fotbal și matricea **R** de mai sus, prin care sunt precizate rezultatele obținute în meciurile directe între echipele care formează grupa.

Astfel,

- Dacă $R[E_1, E_2] = 1$, atunci echipa E_1 a câștigat în fața echipei E2
- Dacă $R[E_1, E_2] = 0$, atunci rezultatul meciului E1-E2 a fost egal
- Dacă $R[E_1, E_2] = -1$, atunci echipa E_1 a pierdut în fața echipei E2

Știind că pentru o victorie se acordă 3 puncte, pentru un egal 1 punct, și pentru o înfrângere nu se acordă niciun punct, determinați numărul de puncte pentru fiecare echipă și prima echipă calificată în faza șaisprezecimilor (echipa cu cele mai multe puncte).

drd. Briciu Anamaria
26 noiembrie 2022

REZOLVARE - C++

```
#include <iostream>

using namespace std;

int main()
{
    int R[4][4]=
    {
        {-2,1,0,1},
        {-1,-2,-1,0},
        {0,1,-2,-1},
        {-1,0,1,-2}
    };

    //char echipe[4][15]
    const char* echipe[4]
        = { "Spania", "Costa Rica", "Germania", "Japonia" };

    int n, max_p=-1, max_e = -1, puncte[4]= {0,0,0,0};
    for (int i=0; i<4; i++)
        for (int j=0; j<4; j++)
        {
            if (R[i][j]==1)
            {
                puncte[i]+=3;
            }
            if (R[i][j]==0)
            {
                puncte[i]+=1;
            }
        }
    for (int i=0; i<4; i++)
    {
        if (puncte[i]>max_p)
        {
            max_p = puncte[i];
            max_e = i;
        }
        cout<<"Pentru echipa "<<echipe[i]<<" punctajul este:"<<puncte[i]<<endl;
    }

    cout<<"Echipa cu cel mai mare punctaj este "<<echipe[max_e]<<" ("<<max_p<<"
                      puncte)."<<endl;

    return 0;
}
```

drd. Briciu Anamaria
26 noiembrie 2022

REZOLVARE - PASCAL

```
program ex1_worldcup;
const
     r:array[0..3,0..3] of integer = ((-2,1,0,1),
                                      (-1,-2,-1,0),
                                      (0,1,-2,-1),
                                      (-1,0,1,-2));
     echipe:array[0..3] of array[0..15] of char = ('Spania',
                                                   'Costa Rica',
                                                   'Germania',
                                                   'Japonia');
     N: integer = 4;
var
   puncte: array[0..3] of integer;
   i,j,max_e,max_p: integer;
begin
     writeln('Solutia in Pascal');
     max_p:=-1;
     max_e:=-1;
     for i:=0 to N-1 do
     begin
          puncte[i]:=0;
     end;
     for i:=0 to N-1 do
         for j:=0 to N-1 do
         begin
             if r[i][j]=1 then
                puncte[i]:=puncte[i]+3;
             if r[i][j]=0 then
                  puncte[i]:=puncte[i]+1;
         end;
     for i:=0 to N-1 do
     begin
          if puncte[i]>max_p then
          begin
               max_p:= puncte[i];
               max_e:=i;
          end;
          writeln('Pentru echipa ', echipe[i],' punctajul este: ', puncte[i]);
     end;
     writeln;
     writeln('Echipa cu cel mai mare punctaj este ', echipe[max_e], '(', max_p,
' puncte).');
     readln;
end.
```

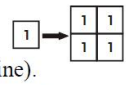
drd. Briciu Anamaria
26 noiembrie 2022

2. [Concurs Admitere UBB (nivel licență), iulie 2017] Prelucrare imagine

Prelucrări imagine

O imagine alb-negru este reprezentată codificat printr-un tablou bidimensional cu valori 0 (pixel alb) și 1 (pixel negru). Asupra imaginii se pot efectua transformări precum:

- Inversarea (I), adică valorile 0 se transformă în 1 și valorile 1 se transformă în 0;
- Rotirea cu 90 de grade în sensul acelor de ceasornic (R);
- Zoom (Z), adică fiecare pixel va fi expandat în 4 pixeli identici cu cel inițial.



O secvență de transformări se definește ca o succesiune de litere I, R și Z (în orice ordine).

Scrieți un program care, fiind dat un tablou bidimensional ***imagine*** având ***m*** linii și ***m*** coloane (***m*** - număr natural, $2 \leq m \leq 10$) și o secvență ***s*** care conține cel mult cinci transformări, aplică aceste transformări și afișează imaginea obținută în urma transformărilor.

Exemplu: dacă ***m*** = 3, ***imagine*** = $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ și ***s*** = (R, I, R, Z), atunci rezultatul va fi $\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$.

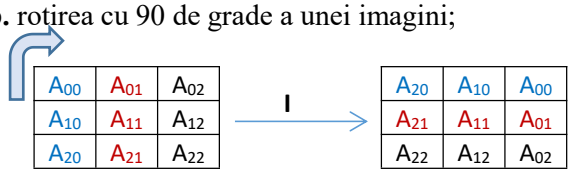
Imaginile intermediare (după aplicarea succesivă a transformărilor R, I, R, Z) sunt:

$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$.

În rezolvare folosiți subprograme pentru:

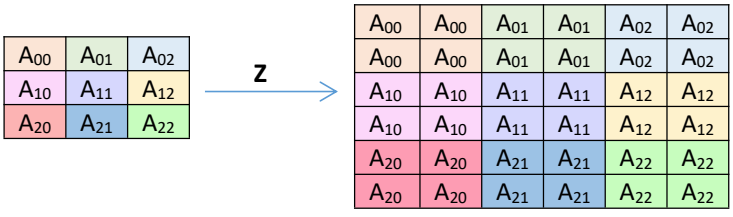
a. citirea datelor de intrare de la tastatură (datele de intrare se consideră corecte în raport cu cerințele);

b. rotirea cu 90 de grade a unei imagini;



c. inversarea unei imagini;

d. aplicarea operației de zoom pe o imagine;



Care este dimensiunea noii matrici?

2*m x 2*m

drd. Briciu Anamaria
26 noiembrie 2022

e. afișarea pe ecran a unei imagini.

Consultații Admitere 2020 - Matrici+imagini (exemplu detaliat)

REZOLVARE - C++

```
#include <iostream>
#include <cstring>
using namespace std;

void citire_matrice(int matrice[][320],int m)
{
    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
            cin>>matrice[i][j];
}

void afisare_matrice(int matrice[][320], int m)
{
    for (int i=0; i<m; i++)
    {
        for (int j=0; j<m; j++)
            cout<<matrice[i][j]<<" ";
        cout<<endl;
    }

}
void rotire_matrice(int matrice[][320], int m)
{
    int aux[320][320];
    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
            aux[i][j]=matrice[m-j-1][i];

    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
            matrice[i][j]=aux[i][j];
}

void invesare_matrice(int matrice[][320], int m)
{

    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
            matrice[i][j]=1-matrice[i][j];
}

void zoom_matrice(int matrice[][320], int& m)
{
    int matrice_z[2*m][2*m];
    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
        {
            matrice_z[2*i][2*j]=matrice[i][j];
            matrice_z[2*i+1][2*j]=matrice[i][j];
            matrice_z[2*i][2*j+1]=matrice[i][j];
            matrice_z[2*i+1][2*j+1]=matrice[i][j];
        }
    m=m*2;
    for (int i=0; i<m; i++)
        for (int j=0; j<m; j++)
            matrice[i][j]=matrice_z[i][j];
}
```

```
int main()
{
    int m, a[320][320];
    char cmd[6];
    cout<<"Dimensiunea matricii:";
    cin>>m;
    citire_matrice(a, m);
    cout<<"Matricea citita este:\n";
    afisare_matrice(a,m);
    cout<<"Transformari:"<<endl;
    cin>>cmd;
    for (int i=0; i<strlen(cmd); i++)
    {
        if (cmd[i]=='R')
            rotire_matrice(a, m);
        if (cmd[i]=='I')
            invesare_matrice(a, m);
        if (cmd[i]=='Z')
            zoom_matrice(a, m);
    }

    cout<<endl;
    afisare_matrice(a,m);
    return 0;
}
```

Observații:

- **cmd[6]** - putem avea maxim 5 transformări (pozițiile 0-4), dar în C++ trebuie să luăm în considerare și caracterul '\0' (null-terminating character, poziția 5)

- **int matrice[][320] vs matrice[320][320]** ca parametru formal
  - Transmitere array ca parametru într-o funcție prin valoare → conversie la pointer („decay")
    - myFunction(int *param), myFunction(int param[10]), myFunction(int param[]) → oricare dintre aceste declarații indică faptul că funcția va primi ca parametru un pointer la int
  - La transmiterea unui tablou bidimensional prin valoare într-o funcție → pointer la array unidimensional (nice explanation with examples of pointers to arrays) → în acest caz, pointer la prima linie din matrice
  - Trebuie să specificăm doar numărul de coloane, pentru a ști unde găsim în memorie următoarea linie (peste 320*sizeof(int) bytes).

- Matricea are maxim dimensiunea 320x320 pentru că singura transformare care îi modifică dimensiunile este cea de Zoom - care îi dublează dimensiunile inițiale; putem avea maxim 5 transformări Zoom. Dacă matricea inițială are dimensiunile maxime de 10x10, atunci prin aplicarea 5xZoom, vom avea 2*(2*(2*(2*(2*10) = $2^5 \cdot 10 = 320$

REZOLVARE - PASCAL

```
program prelucrare_imagine;
type
    matrice = array[0..319,0..319] of integer;
    char_array =  array[0..5] of char;
var
   a: matrice;
   cmd: char_array;
   m, i:integer;

procedure afisare_matrice(var a: matrice; n: integer);
var
   i,j:integer;
begin
     for i:=0 to n-1 do
     begin
          for j:=0 to n-1 do
              write(a[i,j],' ');
          writeln;
     end;
end;

procedure citire_matrice(var a: matrice; n: integer);
var
   i, j: integer;
begin
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             readln(a[i,j]);
end;

procedure rotire_matrice(var a: matrice; n:integer);
var
   i, j: integer;
   aux: matrice;
begin
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             aux[i,j]:= a[n-j-1][i];
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             a[i,j]:= aux[i,j];
end;

procedure inversare_matrice(var a: matrice; n: integer);
var
   i, j: integer;
begin
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             a[i,j]:= 1-a[i,j];
end;
```

drd. Briciu Anamaria
26 noiembrie 2022

```
procedure zoom_matrice(var a:matrice; var n: integer);
var
   i, j: integer;
   az: matrice;
begin
     for i:=0 to n-1 do
         for j:=0 to n-1 do
         begin
              az[2*i,2*j]:= a[i,j];
              az[2*i+1,2*j]:= a[i,j];
              az[2*i,2*j+1]:= a[i,j];
              az[2*i+1,2*j+1]:= a[i,j];
         end;
     n:= n*2;
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             a[i,j]:= az[i,j];
end;

function no_characters(msg:char_array):integer;
var
       i:integer;
begin
       i:=0;
       while ord(msg[i])<>0 do
       begin
              i:=i+1;
       end;
       no_characters:=i;
end;

begin
     writeln('Dimensiunea matricii:');
     read(m);
     citire_matrice(a,m);
     writeln('Matricea citita este:');
     afisare_matrice(a,m);
     writeln('Transformari:');
     read(cmd);
     for i:=0 to no_characters(cmd)-1 do
     begin
         if cmd[i]='R' then
            rotire_matrice(a,m);
         if cmd[i]='I' then
            inversare_matrice(a,m);
         if cmd[i]='Z' then
            zoom_matrice(a,m);
     end;
     writeln('Matricea rezultat este:');
     afisare_matrice(a,m);
     readln;
     readln;
end.
```

drd. Briciu Anamaria
26 noiembrie 2022

3. [Concurs Mate-Info UBB - 2018] Roboțel plimbăreț

B.3. Roboțel plimbăreț (30 puncte)

Un roboțel se poate plimba pe o hartă dată sub forma unei matrice pătratice de dimensiune impară, lăsând în fiecare celulă a hărții un anumit număr de obiecte. Plimbarea roboțelului se desfășoară după următoarele reguli:

- în celula din care pornește roboțelul lasă un obiect, în a doua celulă în care ajunge lasă 2 obiecte, în a treia celulă în care ajunge lasă 3 obiecte, ș.a.m.d.;
- roboțelul pornește din mijlocul ultimei coloane și merge întotdeauna un pas pe diagonală în celula alăturată de sus-dreapta (direcție paralelă cu diagonala secundară) dacă această celulă există și ea este liberă; dacă celula nu există, atunci:
  - dacă roboțelul se află pe ultima coloană, atunci el "sare" în celula aflată pe coloana întâi și linia de deasupra lui dacă aceasta este liberă;
  - dacă roboțelul se află pe prima linie, el "sare" în celula aflată pe ultima linie și coloana din dreapta lui dacă aceasta este liberă;
  - dacă roboțelul se află în colțul dreapta-sus al hărții, el "sare" în celula aflată pe ultima linie și prima coloană dacă aceasta este liberă.
- în situația în care celula pe care trebuie să ajungă nu este liberă, roboțelul face un pas la stânga în celula alăturată de pe aceeași linie cu el.

Aceste reguli asigură vizitarea o singură dată a tuturor celulelor din hartă (și, implicit, evitarea blocajelor în deplasare). După ce roboțelul lasă obiecte în toate celulele hărții, el se oprește.

De exemplu, pentru o hartă cu 5 × 5 celule, primii 22 pași efectuați de roboțel ar fi:

9	3	22	16	15
2	21	20	14	8
	19	13	7	1
18	12	6	5	
11	10	4		17

Scrieți un subalgoritm care determină numărul de obiecte ***nr*** lăsate de roboțel în celulele de pe diagonala principală a hărții. Parametrul de intrare al subalgoritmului este dimensiunea hărții ***n*** (***n*** - număr natural impar, $3 \leq n \leq 100$), iar ***nr*** va fi parametrul de ieșire (***nr*** - număr natural).

Exemplu 1: dacă ***n*** = 5, atunci ***nr*** = 65.
Exemplu 2: dacă ***n*** = 11, atunci ***nr*** = 671.

REZOLVARE 1: Simularea drumului roboțelului (25/30 puncte)

drd. Briciu Anamaria
26 noiembrie 2022

REZOLVARE - C++

```
#include <iostream>
#include <cstring>
using namespace std;

void robot_walk(int harta[][100],int n)
{
    int i, j,initial_i, initial_j, no_obj, covered_cells;

    harta[n/2][n-1] = 1;
    covered_cells = 1;
    i = n/2;
    j = n-1;
    no_obj = 1;
    while (covered_cells<n*n)
    {
        no_obj++;

        initial_i = i;
        initial_j = j;

        if (i>0 && j<=n-2)
        {
            i=i-1;
            j=j+1;
        }

        else if (j==n-1 && i>=1)
        {
            i = i-1;
            j = 0;
        }
        else if (i==0 && j<=n-2)
        {
            i = n-1;
            j = j+1;
        }
        else if (i==0 && j==n-1)
        {
            i=n-1;
            j=0;
        }
        if (harta[i][j]==0)
        {
            harta[i][j] = no_obj;
        }
        else
        {
            i = initial_i;
            j = initial_j-1;
            harta[i][j] = no_obj;
        }
        covered_cells++;


    }
}
```

drd. Briciu Anamaria
26 noiembrie 2022

```
void print_matrix(int matrix[][100], int m)
{
    int i, j;
    for (i=0; i<m; i++)
    {
        for (j=0; j<m; j++)
            cout<<matrix[i][j]<<" ";
        cout<<"\n";
    }
}
int diagonal_sum(int harta[][100],int n)
{
    int diag_sum = 0;
    for (int i=0; i<n; i++)
    {
        diag_sum+=harta[i][i];
    }
    return diag_sum;
}

int main()
{
    int n, harta[100][100];
    cout<<"Dimensiunea hartii:";
    cin>>n;
    robot_walk(harta, n);
    cout<<"Dupa efectuarea tuturor pasilor, harta arata in felul
       urmator:"<<endl;
    print_matrix(harta, n);
    cout<<"Suma de pe diagonala principala este:"<<diagonal_sum(harta, n)<<endl;
}
```

REZOLVARE - PASCAL

```
program walking_robot;
type
	harta = array[0..99,0..99] of integer;
var
	h: harta;
	n:integer;

procedure walk(var m: harta; n:integer);
var
	i,j, initial_i, initial_j,no_obj, covered_cells:integer;
begin
	m[n div 2,n-1]:= 1;
	covered_cells:=1;
	i:= n div 2;
	j:= n-1;
	no_obj:=1;

	while covered_cells<n*n do
	begin
		no_obj:= no_obj+1;
		initial_i:=i;
		initial_j:=j;
		if (i>0) and (j<=n-2) then
		begin
			i:=i-1;
			j:=j+1;
		end
		else if (j=n-1) and (i>=1) then
		begin
			i:=i-1;
			j:=0;
		end
		else if (i=0) and (j<=n-2) then
		begin
			i:=n-1;
			j:=j+1;
		end
		else if (i=0) and (j=n-1) then
		begin
			i:=n-1;
			j:=0;
		end;

		if m[i,j]=0 then
		begin
			m[i,j]:=no_obj;
		end
		else
		begin
			i:= initial_i;
			j:= initial_j-1;
			m[i,j]:=no_obj;
		end;
		covered_cells:=covered_cells+1;
	end;
end;
```

drd. Briciu Anamaria
26 noiembrie 2022

```
procedure afisare_matrice(var a: harta; n: integer);
var
   i,j:integer;
begin
     for i:=0 to n-1 do
     begin
          for j:=0 to n-1 do
              write(a[i,j],' ');
          writeln;
     end;
end;

function diagonal_sum(var a: harta; n:integer):integer;
var
       i,j, diag_sum:integer;
begin
       diag_sum:=0;
       for i:=0 to n-1 do
              diag_sum:=diag_sum+a[i,i];
       diagonal_sum:=diag_sum;
end;

begin
       writeln('Dimensiunea hartii');
       readln(n);
       walk(h, n);
       writeln('Harta dupa efectuarea tuturor pasilor:');
       afisare_matrice(h,n);
       writeln('Suma de pe diagonala principala este ', diagonal_sum(h,n));
       readln;
end.
```

REZOLVARE 2: Găsirea formulei de calcul pentru suma dată (30/30 puncte)

B. 3. Roboțel plimbăreț ..... 30 puncte

- determinarea corectă a valorii prin calcule (***n*n*n+n)/2*** ..... 30 puncte

9	3			
2				8
		13	7	1
	12	6	**5**	
11	10	4		

17	23			
24				18
		13	19	25
	14	20	21	
15	16	22		

		22	16	15
	21	20	14	
25	19	13		
18				24
			23	17

9	3	22	16	15
2	21	20	14	8
25	19	13	7	1
18	12	6	5	
11	10	4		17

Traseul roboțelului este simetric față de centrul pătratului. Altfel spus, centrul pătratului este întotdeauna vizitat exact la mijlocul drumului roboțelului (după (n*n-1)/2 mutări), iar a doua jumătate a drumului este obținută luând prima jumătate, inversând sensul de parcurgere și rotindu-l 180 grade față de centrul pătratului (de exemplu, ultimul pătrățel este mijlocul laturii din stânga). De aici rezultă că, pentru orice pereche de pătrățele simetrice față de centru, suma valorilor lor este n*n+1. Aplicând observația anterioara asupra elementelor de pe diagonală și grupându-le două câte două, rezultă formula sumă=n*(n*n+1)/2

Dacă $n = 5$, suma va fi 65

```
int obiecte(int n){
        return (n * n * n + n) / 2;
}
```

drd. Briciu Anamaria
26 noiembrie 2022

4. Hill Cypher (simplified)

The Hill Cypher („Cifrul Hill") este o metodă de criptare a unui mesaj care folosește matrici. Acesta funcționează în modul următor:

A. Se dă un tabel de corespondență între litere și valori numerice, în mod uzual cel care asociază literei A valoarea 0, literei B valoarea 1, ..., literei Z valoarea 25.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Tabel 1

B. Se precizează o **matrice-cheie**, care are următoarele proprietăți:
   a) este o matrice pătratică de dimensiuni **nxn** inversabilă
   b) Determinantul matricii nu are în comun niciun factor cu baza modulară folosită în criptare (e.g. dacă folosim simbolurile de mai sus, baza este 26, astfel determinantul nu poate fi divizibil cu 2 sau 13)

C. Pentru **criptare**: fiecare bloc de *n* litere este convertit în matrici de dimensiuni **nx1** conform tabelului de la punctul A, apoi se înmulțesc pe rând matricile obținute cu matricea-cheie. Rezultatele (*mod* baza) se interpretează ca șiruri de caractere reprezentând coduri.

D. Pentru **decodare**: având mesajul X dat prin matrici de dimensiuni **nx1**, se înmulțesc pe rând matricile cu matricea inversă[1] matricii cheie. Se interpretează rezultatul cu ajutorul tabelului de corespondență de la punctul A.

Exemplu

Se dă tabelul de corespondență din descrierea de la punctul A.

Se dă următoarea matrice cheie:

6	**24**	**1**
13	**16**	**10**
20	**17**	**15**

Are dimensiunea 3x3 (n=3).

Se dă următorul mesaj:

MATRIXGLITCH

Criptarea mesajului:

Se împarte mesajul MATRIXGLITCH în blocuri de *n* caractere.

n=3

M	A	T	R	I	X	G	L	I	T	C	H

[1] „modular inverse": within modular arithmetic, the modular inverse of A (mod C) is A^{-1}: $(A * A^{-1}) \equiv 1 \pmod{C}$ or equivalently $(A * A^{-1}) \bmod C = 1$

Pentru fiecare bloc, se obține matricea *nx1* aferentă.

M	→	12
A		0
T		19

R	→	17
I		8
X		23

G	→	6
L		11
I		8

T	→	19
C		2
H		7

Fiecare dintre aceste matrici *nx1* se înmulțește cu matricea-cheie.

6	24	1
13	16	10
20	17	15

× | 12 / 0 / 19 | = | 12*6 + 24*0 + 1*19 / 13*12 + 16*0 + 10*19 / 20*12 + 17*0 + 15*19 | = | 91 / 346 / 525 |

= | 13 / 8 / 5 | (MOD 26)

Astfel, codul pentru MAT este NIF (N=13, I=8, F=5 conform tabelului).

Analog se procedează pentru celelalte 3 blocuri (RIX, GLI, TCH).

MATRIXGLITCH → NIFFHPWWLNLZ

Decriptarea mesajului:

Mesaj de decriptat: NIFFHPWWLNLZ

N	I	F	F	H	P	W	W	L	N	L	Z

Se obțin matricile *nx1* corespunzătoare blocurilor de *n* litere din mesajul de decriptat. Analog cu procesul de criptare, aceste matrici se vor înmulți, însă de data aceasta, cu <u>inversa</u> matricii-cheie.

Inversa (*modular inverse) matricii cheie:

MODULAR INVERSE: Inversul modular al unui număr x (mod C) este x^{-1}:

$$(x \cdot x^{-1}) \bmod C = 1$$

$$\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}^{-1} \pmod{26} \equiv \begin{pmatrix} 8 & 5 & 10 \\ 21 & 8 & 21 \\ 21 & 12 & 8 \end{pmatrix}$$

8	5	10
21	8	21
21	12	8

× | 5 / 7 / 15 | = | 8*5 + 5*7 + 10*15 / 21*5 + 8*7 + 21*15 / 21*5 + 12*7 + 8*15 | = | 225 / 476 / 309 | MOD 26

= | 17 / 8 / 23

drd. Briciu Anamaria
26 noiembrie 2022

Astfel, pentru codul obținut șirul de caractere corespondent este RIX (R=17, I=8, X=23 conform tabelului).

Analog se procedează pentru celelalte 3 blocuri din mesajul de decodat.

Folosind tabelul de corespondență precizat anterior (Tabel 1):

1. Să se implementeze algoritmul de criptare al unui mesaj de cel mult 20 de caractere, format din litere A-Z. Dimensiunea matricii cheie, matricea cheie și mesajul se citesc de la tastatură.

2. Fiind dată matricea inversă a matricii cheie (fie direct în program, fie citită de la tastatură), decodați mesajul **GIEQQALBBEVXOFB**.

REZOLVARE:

Trebuie să scriem subprograme pentru:

- Citirea matricii (fie a matricii-cheie pentru cerința 1, fie a matricii inverse pentru cerința 2)
- Criptarea unui mesaj
  - Transformarea unui mesaj dat ca șir de caractere într-un mesaj numeric, conform Tabel 1
  - Împărțirea mesajului în subșiruri de lungime egală cu dimensiunea matricii cheie
  - Înmulțirea fiecărei părți din mesaj cu matricea cheie
  - Transformarea rezultatelor numerice obținute înapoi în caractere
- Decriptarea unui mesaj
  - Procesul este similar cu procesul de criptare, singura diferență este matricea folosită;
  - Se vor folosi subprogramele definite pentru criptare

REZOLVARE - C++

```
#include <iostream>
#include <math.h>
#include <cstring>
using namespace std;

int key_matrix[10][10], inv_key_matrix[10][10], crt_block[3][1];
char msg[21],msg_d[21], coded_msg[21];
int msg_n[21], msg_dn[21];
int n;

void read_matrix(int matrix[][10],int n)   //get key and message from user
{

    int i, j;
    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            cin>>matrix[i][j];


}

void msg_to_numeric(char message[], int msg_length, int msg_numeric[])
{
    for(int i = 0; i < msg_length; i++)
        msg_numeric[i] = message[i] - 65;
}


void multiply_matrices(int A[][10], int B[][1], int C[][1], int n)
{

    int i, j, k, no_rowsA=n, no_rowsB=n, no_colsA=n, no_colsB=1;

    //Matricea rezultat va avea dimensiunea no_rowsA x no_colsB
    for(i = 0; i < no_rowsA; i++)
        for(j = 0; j < no_colsB; j++)
            C[i][j] = 0;

    for(i = 0; i < no_rowsA; i++)
        for(j = 0; j < no_colsB; j++)
            for(k=0; k < no_colsA; k++)
                C[i][j] += A[i][k] * B[k][j];
}

void encrypt(int key_matrix[][10], int numeric_msg[], int msg_length, int block_size, char
                        coded_msg[])
{
    int crt_msg[block_size][1];
    int encrypted_msg[block_size][1];
    int i=0;
    char c;
    while (i<msg_length)
    {
        for (int j=0; j<block_size; j++)
            crt_msg[j][0]=numeric_msg[i+j];

        multiply_matrices(key_matrix,crt_msg,encrypted_msg,block_size);

        for(int k = 0; k < block_size; k++)
        {
            c = (char)((encrypted_msg[k][0] % 26) + 65);
            strncat(coded_msg,&c,1);
        }

        i+=block_size;

    }
}
```

drd. Briciu Anamaria
26 noiembrie 2022

```
void run(char matrix_type[], char mode[])
{
    cout<<"Reading "<<matrix_type<<" matrix..."<<endl;
    cout<<"Size of "<<matrix_type<<" matrix:\n";
    cin>>n;
    cout<<"The "<<matrix_type<<" matrix is:\n";
    read_matrix(key_matrix,n);

    cout<<"Message to "<<mode<<" (use letters A-Z):\n";
    cin>>msg;

    msg_to_numeric(msg, msg_n);
    encrypt(key_matrix, msg_n, strlen(msg), n, coded_msg);
    cout<<"The "<<mode<<"ed message is:"<<coded_msg<<endl;
}

int main()
{

    char menu_string[100] ="1. Encrypt message\n2. Decrypt message\n3. Exit\n";
    int option;
    while (1)
    {
        strcpy(coded_msg, "");
        cout<<menu_string;
        cout<<"Option is:";
        cin>>option;
        if (option==1)
            run("key", "encrypt");
        if (option==2)
            run("inverse", "decrypt");
        if (option==3)
            break;

    }



}
```

drd. Briciu Anamaria
26 noiembrie 2022

Explicații:

- Subprogramul **read_matrix**
  - citește elementele unei matrici de dimensiune *nxn* de la tastatură
  - poate fi folosit atât pentru citirea matricii cheie, cât și pentru citirea matricii inverse

- Subprogramul **msg_to_numeric**
  - Transformă mesajul dat ca șir de caractere într-un șir de numere naturale
  - Se folosește codul ASCII
  - Literele A-Z au coduri consecutive între 65 și 90, astfel prin scăderea valorii 65 vom obține valori între 0-25
  - **limitare** a soluției în forma curentă: nu vom putea introduce decât mesaje formate din literele A-Z/litere mari; **adăugarea** unor noi caractere pentru a fi folosite în mesaje se poate face prin asignarea de coduri acestora - considerăm că numărul 26 corespunde caracterului spațiu, numărul 27 caracterului ‘, etc -> se va schimba baza modulară

- Subprogramul **multiply_matrices**
  - Înmulțește două matrici A, B date ca parametru și returnează matricea obținută ca rezultat prin parametrul C
  - Se specifică numărul de linii și numărul de coloane pentru fiecare matrice; se presupune că numărul de coloane al matricii A este egal cu numărul de linii al matricii B pentru a se putea efectua înmulțirea
  - Matricea A în această problemă va fi de dimensiune *nxn* (matricea cheie sau inversa matricii), iar matricea B de dimensiuni *nx1* (matricile care reprezintă blocuri de *n* caractere din mesaj)

- Subprogramul **encrypt**
  - Are ca parametri
    - matricea cu care se înmulțește **(key_matrix**) - poate fi matricea cheie sau matricea inversă
    - Mesajul de criptat sau decriptat în format numeric (**numeric_msg**) și lungimea acestuia (**msg_length**)
    - Dimensiunea matricii/a unui bloc de caractere (**block_size = n**)
    - **coded_msg**, prin care se returnează mesajul criptat/decriptat
  - Pentru fiecare bloc de *n* caractere din mesaj, se aplică modul de criptare Hill, descris anterior: înmulțirea matricilor, preluarea rezultatului mod 26, transformarea valorilor numerice în caractere, construirea mesajului criptat/decriptat

O altă **limitare** a soluției în forma curentă: nu s-a luat în considerare niciun caracter care semnifică absența unei litere (e.g. spațiu sau -); pentru un rezultat interpretabil va trebui să oferim ca date de intrare șiruri de caractere care au lungimea multiplu de *n*. **Soluția** este adăugarea unor astfel de caractere în tabelul de corespondență.

Pentru ușurința utilizării, programul are un meniu prin care se poate alege între a cripta sau a decripta un mesaj.

EXTENSIONS:
- Adăugarea altor caractere pentru formarea mesajului
- Implementarea unui subprogram care calculează matricea inversă matricii cheie în mod automat

REZOLVARE - Pascal

```
program hill_cypher;
type
       matrice_p = array[0..9, 0..9] of integer;
       matrice_col = array[0..9, 0..1] of integer;
       array_num = array[0..20] of integer;
       array_char = array[0..20] of char;
var
       matrix: matrice_p;
       msg, coded_msg: array_char;
       msg_n:array_num;
       n, option:integer;

function no_characters(msg:array_char):integer;
var
       i:integer;
begin
       i:=0;
       while ord(msg[i])<>0 do
       begin
              i:=i+1;
       end;
       no_characters:=i;
end;

procedure read_matrix(var a: matrice_p; n: integer);
var
   i, j: integer;
begin
     for i:=0 to n-1 do
         for j:=0 to n-1 do
             readln(a[i,j]);
end;

procedure msg_to_numeric(msg: array_char; msg_length: integer; var
msg_numeric:array_num);
var
       i: integer;
begin
       for i:=0 to msg_length-1 do
              msg_numeric[i]:=ord(msg[i]) - 65;
end;
```

drd. Briciu Anamaria
26 noiembrie 2022

```
procedure multiply_matrices(A: matrice_p; B: matrice_col; var C:matrice_col;
n:integer);
var
	i, j, k, rowsA, rowsB, colsA, colsB:integer;
begin
	rowsA:=n;
	rowsB:=n;
	colsA:=n;
	colsB:=1;
	for i:=0 to rowsA-1 do
		for j:=0 to colsB-1 do
			C[i,j]:=0;

	for i:=0 to rowsA-1 do
		for j:=0 to colsB-1 do
			for k:=0 to colsA-1 do
				C[i,j]:= C[i,j] + (A[i,k]*B[k,j]);

end;

procedure process_message(m: matrice_p; numeric_msg: array_num; msg_length:
integer; block_size:integer; var coded_msg:array_char);
var
	crt_msg, encrypted_msg:matrice_col;
	i,j,k:integer;
	c:char;
begin
	i:=0;
	while (i<msg_length) do
	begin
		for j:=0 to block_size-1 do
			crt_msg[j,0]:=numeric_msg[i+j];
		multiply_matrices(m,crt_msg,encrypted_msg,block_size);
		for k:=0 to block_size-1 do
		begin
			c:= chr((encrypted_msg[k,0] mod 26)+65);
			coded_msg[i+k]:=c;
		end;
		i:=i+block_size;
	end;
end;
```

drd. Briciu Anamaria
26 noiembrie 2022

```
procedure run(matrix_type: array_char; mode: array_char);
begin
	writeln('Reading ', matrix_type, ' matrix...');
	writeln('Size of ', matrix_type, ' matrix:');
	readln(n);
	writeln('The ', matrix_type, ' matrix is:');
	read_matrix(matrix, n);
	writeln('Message to ', mode, ' (use letters A-Z):');
	readln(msg);
	msg_to_numeric(msg, no_characters(msg), msg_n);
	process_message(matrix, msg_n, no_characters(msg), n, coded_msg);
	writeln('The ', mode, 'ed message is: ', coded_msg);
end;

procedure show_menu();
begin
	writeln('1. Encrypt message');
	writeln('2. Decrypt message');
	writeln('3. Exit');
end;

begin
	while true do
	begin
		show_menu();
		writeln('Option is:');
		readln(option);
		if option=1 then
			run('key', 'encrypt');
		if option=2 then
			run('inverse', 'decrypt');
		if option=3 then
			break;
	end;
end.
```

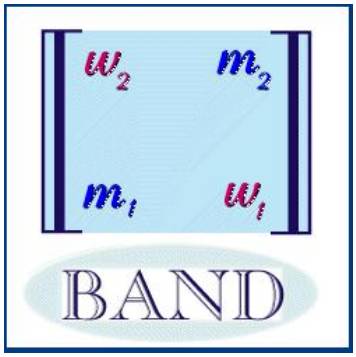
MATRICES...EVERYWHERE

Pentru exercițiu cu tablouri bidimensionale, din subiectele de pe pagina Subiecte ani precedenți puteți rezolva următoarele probleme:

5. [Concurs Mate-Info UBB - 2014] Subiectul III
6. [Concurs Admitere UBB (nivel licență), septembrie 2014] Subiectul III
7. [Concurs Mate-Info UBB - 2015] Subiectul III
8. [Concurs Admitere UBB (nivel licență), iulie 2015] Subiectul Ia
9. [Concurs Admitere UBB (nivel licență), iulie 2015] Subiectul III

…including the real world

Did you know: DANCING MATHEMATICS



In contra-dances, dancers form groups of four (two couples). Each square block, consisting of two couples, can be thought of as a two-by-two matrix. Matrices can describe the *possible arrangements of dancers in a group of four after various figures.*

More abstractly*, changes to these arrangements can be thought of as geometric transformations.* Most figures executed by each group of four can be seen as reflections, rotations, and translations.

Contact: anamaria.briciu@ubbcluj.ro

drd. Briciu Anamaria
26 noiembrie 2022