

CATEDRA

FIZICĂ, MATEMATICĂ ȘI TEHNOLOGII INFORMAȚIONALE

SPECIALITATEA

TEHNOLOGII INFORMAȚIONALE ÎN INSTRUIRE

Structuri algebrice pe calculator

Laboratorul Nr.3

Realizat: Cojucovschi Ion

Grupa: CIII

Verificat: Chiriac Liubomir

Chișinău, 2018

Algoritmi de realizare a produsului direct dintre doi grupoizi

Problemă.Se cere de elaborat un algoritm care efectuează produsul cartezian dintre grupoidul (G, \cdot) și $(H, *)$ conform următoarei reguli:

(.

Să se realizeze produsul direct dintre grupoizii indicați mai jos:

Ex.1

\cdot	1	2
1	1	2
2	2	1

$*$	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

Ex.2

\cdot	1	2	3	4
1	2	3	1	4
2	4	1	3	2
3	3	2	4	1
4	1	4	2	3

$*$	1	2	3	4	5	6	7
1	1	4	7	3	6	2	5
2	6	2	5	1	4	7	3
3	4	7	3	6	2	5	1
4	2	5	1	4	7	3	6
5	7	3	6	2	5	1	4
6	5	1	4	7	3	6	2
7	3	6	2	5	1	4	7

Ex.3

\cdot	1	2	3	4
1	1	2	4	3
2	2	3	1	4
3	4	1	3	2
4	3	4	2	1

$*$	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

Ex.4

\cdot	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

$*$	1	2	3	4
1	1	4	2	3
2	3	2	4	1
3	4	1	3	2
4	2	3	1	4

Ex.5

\cdot	1	2	3	4	5	6	7
1	1	4	7	3	6	2	5
2	6	2	5	1	4	7	3
3	4	7	3	6	2	5	1
4	2	5	1	4	7	3	6
5	7	3	6	2	5	1	4
6	5	1	4	7	3	6	2
7	3	6	2	5	1	4	7

$*$	1	2	3
1	1	2	3
2	3	1	2
3	2	3	1

Ex.6

\cdot	1	2	3	4
---------	---	---	---	---

$*$	1	2	3	4
-----	---	---	---	---

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

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

Ex.7

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

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

Ex.8

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

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

Elaborarea programului:

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAboratorul_numarul_3

{

    class Program

    {

        static int[,] a, b, rez, t1, p1, gr;

        static int[] c, f;

        static int r1, d1, d2, r, d, r2, n, m, t, i, j, p, l;

        static void Main(string[] args)

        {

            t1 = new int[30, 30];
```

```

int[,] b = new int[4, 4] { { 0,0,0,0},{0,1,2,3 },{0,3,1,2 },{0,2,3,1 }
};

int[,] ai = new int[3, 3] { { 0,0,0},{0,1,2 },{0,2,1 } };

int n = 0;

Console.Write("Dati ordinul matricei a, n=");

n = Convert.ToInt32(Console.ReadLine());

Console.Write("Dati ordinul matricei b, m=");

m = Convert.ToInt32(Console.ReadLine());

////initializationmatrix(a,n);      ///initiate matrix

produs_cartezian(out a, ai, b, n, m);

n = n * m;

r = 0; r1 = 0;

afisare(a, n);

asociativ(a, n);

medial(a, n);

paramedial(a, n);

bicomutativ(a, n);

ag_gr(a, n);

ga_gr(a, n);

ga_gr1(a, n);

ad_gr(a, n);

da_gr(a, n);

hexagonal(a, n);

dist_dr(a, n);

dist_st(a, n);

unitate_dreapta(a, n, out r);

unitate_stanga(a, n, out r1);

unitate(ref r, ref r1);

ward(a, n);

ward_invers(a, n);

```

```
Console.ReadKey();
```

```
}
```

```
public static void initializationmatrix(out int[,] a,int n) {
```

```
    ///initializam rarrayul
```

```
    a = new int[15, 15];
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        for (int j = 0; j < n; j++)
```

```
        {
```

```
            Console.WriteLine("a[" + i + 1 + "," + j + 1 + "]=");
```

```
            a[i, j] = Convert.ToInt32(Console.ReadLine());
```

```
        }
```

```
    }
```

```
}
```

```
public static void produs_cartezipan(out int[,] masiv,int[,] a,int[,] b,int  
n,int m)
```

```
{    int k = 1;
```

```
    masiv = new int[30, 30];
```

```
    int[] c= new int[100];
```

```
    int[] f= new int[100];
```

```
    for (int i = 1; i < n + 1; i++)
```

```
        for (int j = 1; j < m + 1; j++)
```

```
        {
```

```

        f[k] = j;

        c[k] = i;

        t1[i,j] = k;

        k=k+1;

    }

    for (int i = 1; i < m * n+1; i++)

        for (int j = 1; j < n * m+1; j++)

            {

                masiv[i, j] = t1[a[c[i], c[j]], b[f[i], f[j]]];

            }

}

```

```

public static void afisare(int[,] a, int n)

{

    for (int i = 1; i < n + 1; i++)

    {

        for (int j = 1; j < n + 1; j++)

        {

            Console.Write(" "+a[i, j]);

        }

        Console.WriteLine("//"+i+"//"+j);

    }

    Console.WriteLine();

}

```

```

public static void asociativ(int[,] masiv, int n)

{

```

```

int l = 0;

for (int i = 1; i < n + 1; i++)
{
    for (int j = 1; j < n + 1; j++)
    {
        for (int k = 1; k < n + 1; k++)
        {
            d = masiv[j, k];
            d1 = masiv[i, j];
            if (masiv[i, d] != masiv[d1, k]) l++;
        }
    }
}

if (l == 0) Console.WriteLine("ESTE ASOCIATIV");
else Console.WriteLine("NU ESTE ASOCIATIV");
}

```

```

public static void medial(int[,] masiv, int n)
{
    int l = 0;

    for (int i = 1; i < n + 1; i++)
    {
        for (int j = 1; j < n + 1; j++)
        {
            for (int k = 1; k < n + 1; k++)
            {
                for (int t = 1; t < n + 1; t++)
                {

```

```

        d = masiv[i, j];

        r = masiv[k, t];

        d1 = masiv[i, k];

        r1 = masiv[j, t];

        if (masiv[d, r] != masiv[d1, r1])

            l++;

    }

}

}

}

if (l == 0) Console.WriteLine("ESTE MEDIAL");

else Console.WriteLine("NU ESTE MEDIAL");

}

public static void paramedial(int[,] masiv, int n)

{

    int l = 0;

    for (int i = 1; i < n + 1; i++)

    {

        for (int j = 1; j < n + 1; j++)

        {

            for (int k = 1; k < n + 1; k++)

            {

                for (int t = 1; t < n + 1; t++)

                {

                    d = masiv[i, j];

                    r = masiv[k, t];

                    d1 = masiv[t, j];

                    r1 = masiv[k, i];

                    if (masiv[d, r] != masiv[d1, r1]) l++;

                }

            }

        }

    }

}

```



```

        }

    }

}

if (l == 0) Console.WriteLine("ESTE PARAMEDIAL");

else Console.WriteLine("NU ESTE PARAMEDIAL");

}

public static void bicomutativ(int[,] masiv, int n)

{

    int l = 0;

    for (int i = 1; i < n + 1; i++)

    {

        for (int j = 1; j < n + 1; j++)

        {

            for (int k = 1; k < n + 1; k++)

            {

                for (int t = 1; t < n + 1; t++)

                {

                    d = masiv[i, j];

                    r = masiv[k, t];

                    dl = masiv[t, k];

                    rl = masiv[j, i];

                    if (masiv[d, r] != masiv[dl, rl]) l++;

                }

            }

        }

    }

}

if (l == 0) Console.WriteLine("ESTE BICOMUTATIV");

```

```

        else Console.WriteLine("NU ESTE BICOMUTATIV");
    }

    public static void ag_gr(int[,] masiv, int n)
    {

        int l = 0;

        for (int i = 1; i < n + 1; i++)
        {
            for (int j = 1; j < n + 1; j++)
            {
                for (int k = 1; k < n + 1; k++)
                {
                    d = masiv[i, j];
                    d1 = masiv[k, j];
                    if (masiv[d, k] != masiv[d1, i]) l++;
                }
            }
        }

        if (l == 0) Console.WriteLine("ESTE AG GRUPOID");
        else Console.WriteLine("NU ESTE AG GRUPOID");
    }

```

```

    public static void ga_gr(int[,] masiv, int n)
    {

        int l = 0;

        for (int i = 1; i < n + 1; i++)
        {
            for (int j = 1; j < n + 1; j++)
            {
                for (int k = 1; k < n + 1; k++)
                {

```

```

        d = masiv[i, j];

        d1 = masiv[j, i];

        if (masiv[d, k] != masiv[k, d1]) l++;

    }

}

if (l == 0) Console.WriteLine("ESTE GA GRUPOID");
else Console.WriteLine("NU ESTE GA GRUPOID");
}

```

```

public static void ga_gr1(int[,] masiv, int n)
{
    int l = 0;

    for (int i = 1; i < n + 1; i++)
    {
        for (int j = 1; j < n + 1; j++)
        {
            for (int k = 1; k < n + 1; k++)
            {
                d = masiv[i, j];

                d1 = masiv[k, i];

                if (masiv[d, k] != masiv[d1, j]) l++;

            }

        }

    }

    if (l == 0) Console.WriteLine("ESTE GA1 GRUPOID");
    else Console.WriteLine("NU ESTE GA1 GRUpoid");
}

```

```

public static void ad_gr(int[,] masiv, int n)
{
    int l = 0;

```

```

        for (int i = 1; i < n + 1; i++)
        {
            for (int j = 1; j < n + 1; j++)
            {
                for (int k = 1; k < n + 1; k++)
                {
                    d = masiv[j, k];
                    d1 = masiv[j, i];
                    if (masiv[i, d] != masiv[k, d1]) l++;
                }
            }
        }

        if (l == 0) Console.WriteLine("ESTE AD GRUPOID");
        else Console.WriteLine("NU ESTE AD GRUpoid");
    }
}

```

```

public static void da_gr(int[,] masiv, int n)
{
    int l = 0;
    for (int i = 1; i < n + 1; i++)
    {
        for (int j = 1; j < n + 1; j++)
        {
            for (int k = 1; k < n + 1; k++)
            {
                d = masiv[j, k];
                d1 = masiv[i, j];
                if (masiv[i, d] != masiv[k, d1]) l++;
            }
        }
    }

    if (l == 0) Console.WriteLine("ESTE DA GRUPOID");
}

```

```
else Console.WriteLine("NU ESTE DA GRUpoid");
```

```
}
```

```
public static void hexagonal(int[,] masiv, int n)
```

```
{
```

```
    int l = 0;
```

```
    for (int i = 1; i < n + 1; i++)
```

```
    {
```

```
        for (int j = 1; j < n + 1; j++)
```

```
        {
```

```
            for (int k = 1; k < n + 1; k++)
```

```
            {
```

```
                for (int t = 1; t < n + 1; t++)
```

```
                {
```

```
                    d = masiv[i, j];
```

```
                    r = masiv[k, t];
```

```
                    d1 = masiv[i, k];
```

```
                    r1 = masiv[j, t];
```

```
                    r2 = masiv[j, i];
```

```
                    if (masiv[i, i] != i || masiv[d, r] != masiv[d1, r1] ||  
masiv[i, r2] != masiv[d, i] && masiv[d, i] != j) l++;
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    if (l == 0) Console.WriteLine("ESTE HEXAGONAL");
```

```
    else Console.WriteLine("NU ESTE HEXAGONAL");
```

```
}
```

```

public static void dist_dr(int[,] masiv, int n)
{
    int l = 0;

    for (int i = 1; i < n + 1; i++)
        for (int j = 1; j < n + 1; j++)
            for (int k = 1; k < n + 1; k++)
                {
                    d = masiv[i, j];
                    d1 = masiv[i, k];
                    r1 = masiv[j, k];
                    if (masiv[d, k] != masiv[d1, r1]) l++;
                }

    if (l == 0) Console.WriteLine("ESTE DISTRIBUTIV DE DREAPTA");
    else Console.WriteLine("NU ESTE DISTRIBUTIV DE DREAPTA");
}

```

```

public static void dist_st(int[,] masiv, int n)
{
    int l = 0;

    for (int i = 1; i < n + 1; i++)
        for (int j = 1; j < n + 1; j++)
            for (int k = 1; k < n + 1; k++)
                {
                    d = masiv[i, j];
                    d1 = masiv[k, i];
                    r1 = masiv[k, j];
                    if (masiv[k, d] != masiv[d1, r1]) l++;
                }
}

```

```

        if (l == 0) Console.WriteLine("ESTE DISTRIBUTIV DE STANGA");
        else Console.WriteLine("NU ESTE DISTRIBUTIV DE STANGA");

    }

```

```

public static void unitate_dreapta(int[,] masiv, int n, out int r)
{
    int l;

    int j = 0; r = 0;

    for (int i = 1; i < n + 1; i++)
    {
        l = 0;

        j++;

        if (masiv[j, i] == i)
        {
            for (int k = 1; k < n + 1; k++)
            {
                if (masiv[k, j] == k) l++;

                if (l == n) r = j;
            }
        }
    }

    if (r != 0) Console.WriteLine("ESTE UNITATE DREAPTA " + r);
    else Console.WriteLine("NU ESTE UNITATE STANGA");
}

```

```

public static void unitate_stanga(int[,] masiv, int n, out int r2)
{
    int l;

```

```

int j = 0; r2 = 0;

for (int i = 1; i < n + 1; i++)
{
    l = 0;

    j++;

    if (masiv[i, j] == i)
    {
        for (int k = 1; k < n + 1; k++)
        {
            if (masiv[j, k] == k) l++;

            if (l == n) r2 = j;
        }
    }
}

if (r2 != 0) Console.WriteLine("ESTE UNITATE STANGA " + r2);
else Console.WriteLine("NU ESTE UNITATE STANGA");
}

public static void unitate(ref int r, ref int r2)
{

    if (r1 == r2 && r > 0) Console.WriteLine("Este unitate " + r);
    else Console.WriteLine("NU este unitate");

}

public static void ward(int[,] masiv, int n)
{
    int l = 0;

    for (int i = 1; i < n + 1; i++)
        for (int j = 1; j < n + 1; j++)
            for (int k = 1; k < n + 1; k++)

```



```

    {
        d = masiv[i, j];
        d1 = masiv[i, k];
        d2 = masiv[j, k];
        if (d != masiv[d1, d2]) l += 1;
    }

    if (l == 0) Console.WriteLine("ESTE WARD");
    else Console.WriteLine("NU ESTE WARD");
}

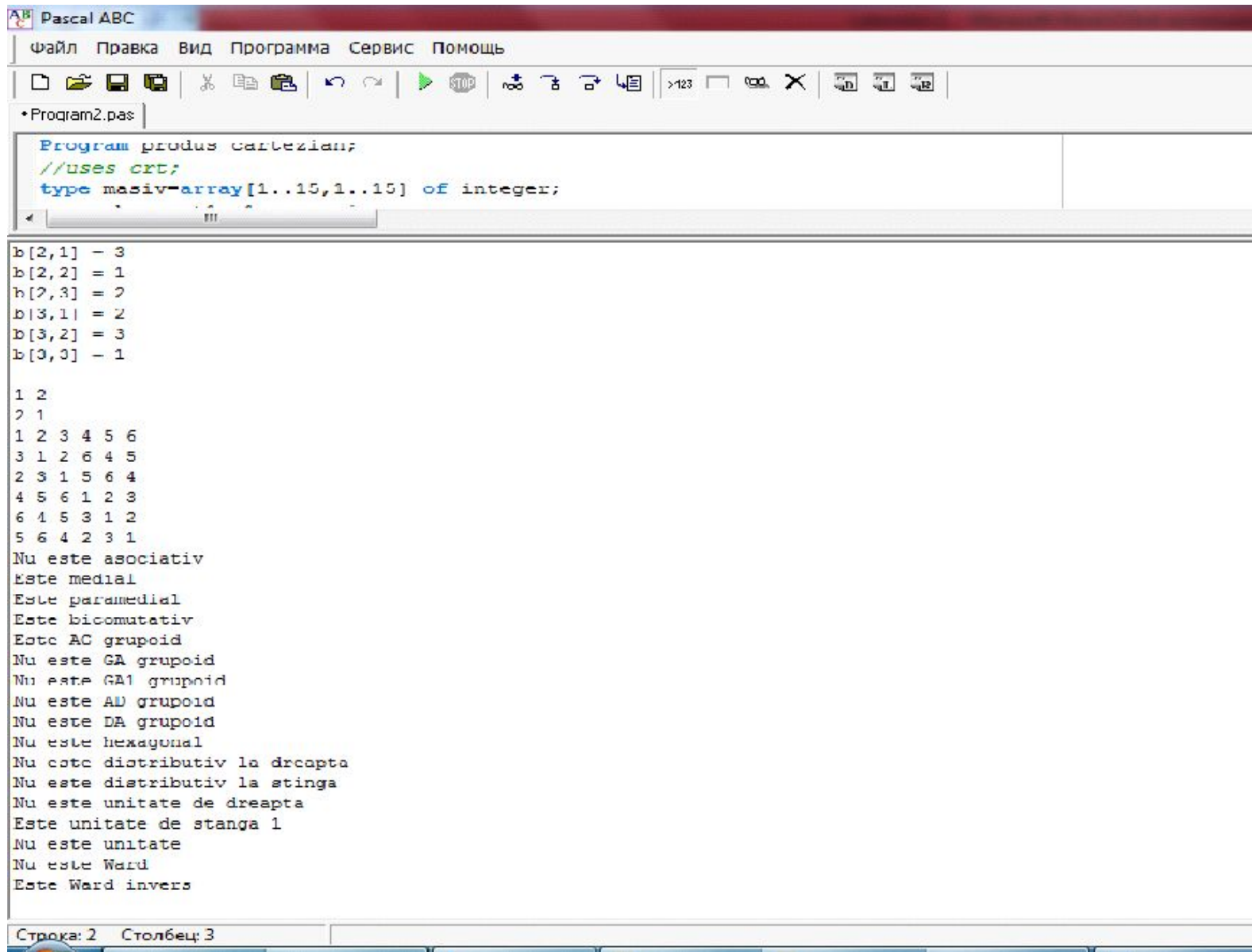
public static void ward_invers(int[,] masiv, int n)
{
    int l = 0;
    for (int i = 1; i < n + 1; i++)
        for (int j = 1; j < n + 1; j++)
            for (int k = 1; k < n + 1; k++)
                {
                    d = masiv[i, j];
                    d1 = masiv[k, i];
                    d2 = masiv[k, j];
                    if (d != masiv[d1, d2]) l += 1;
                }

    if (l == 0) Console.WriteLine("ESTE WARD INVERS");
    else Console.WriteLine("NU ESTE WARD INVERS");
}
}
}

```

În urma compilării sau obținut următoarele rezultate:

Ex. 1



```
Pascal ABC
Файл Правка Вид Программа Сервис Помощь
•Program2.pas
Program produs Cartezian;
//uses crt;
type masiv=array[1..15,1..15] of integer;

b[2,1] = 3
b[2,2] = 1
b[2,3] = 2
b[3,1] = 2
b[3,2] = 3
b[3,3] = 1

1 2
2 1
1 2 3 4 5 6
3 1 2 6 4 5
2 3 1 5 6 4
4 5 6 1 2 3
6 4 5 3 1 2
5 6 4 2 3 1
Nu este asociativ
Este medial
Este paramedial
Este bicomutativ
Este AC grupoid
Nu este GA grupoid
Nu este GA1 grupoid
Nu este AD grupoid
Nu este DA grupoid
Nu este hexagonal
Nu este distributiv la dreapta
Nu este distributiv la stanga
Nu este unitate de dreapta
Este unitate de stanga 1
Nu este unitate
Nu este Ward
Este Ward invers

Строка: 2 Столбец: 3
```

Ex. 2

2314
4132
3241
1423
81114101391215182117201619147362522252824272326
13912811141020161915182117625147327232622252824
11141013912818211720161915473625125282427232622
91281114101316191518211720251473623262225282427
14101391281121172016191518736251428242723262225
12811141013919151821172016514736226222528242723
10139128111417201619151821362514724272326222528
22252824272326147362515182117201619811141013912
27232622252824625147320161915182117139128111410
25282427232622473625118211720161915111410139128
23262225282427251473616191518211720912811141013

28 24 27 23 26 22 25 7 3 6 2 5 1 4 21 17 20 16 19 15 18 14 10 13 9 12 8 11
 26 22 25 28 24 27 23 5 1 4 7 3 6 2 19 15 18 21 17 20 16 12 8 11 14 10 13 9
 24 27 23 26 22 25 28 3 6 2 5 1 4 7 17 20 16 19 15 18 21 10 13 9 12 8 11 14
 15 18 21 17 20 16 19 8 11 14 10 13 9 12 22 25 28 24 27 23 26 1 4 7 3 6 2 5
 20 16 19 15 18 21 17 13 9 12 8 11 14 10 27 23 26 22 25 28 24 6 2 5 1 4 7 3
 18 21 17 20 16 19 15 11 14 10 13 9 12 8 25 28 24 27 23 26 22 4 7 3 6 2 5 1
 16 19 15 18 21 17 20 9 12 8 11 14 10 13 23 26 22 25 28 24 27 2 5 1 4 7 3 6
 21 17 20 16 19 15 18 14 10 13 9 12 8 11 28 24 27 23 26 22 25 7 3 6 2 5 1 4
 19 15 18 21 17 20 16 12 8 11 14 10 13 9 26 22 25 28 24 27 23 5 1 4 7 3 6 2
 17 20 16 19 15 18 21 10 13 9 12 8 11 14 24 27 23 26 22 25 28 3 6 2 5 1 4 7
 1 4 7 3 6 2 5 22 25 28 24 27 23 26 8 11 14 10 13 9 12 15 18 21 17 20 16 19
 6 2 5 1 4 7 3 27 23 26 22 25 28 24 13 9 12 8 11 14 10 20 16 19 15 18 21 17
 4 7 3 6 2 5 1 25 28 24 27 23 26 22 11 14 10 13 9 12 8 18 21 17 20 16 19 15
 2 5 1 4 7 3 6 23 26 22 25 28 24 27 9 12 8 11 14 10 13 16 19 15 18 21 17 20
 7 3 6 2 5 1 4 28 24 27 23 26 22 25 14 10 13 9 12 8 11 21 17 20 16 19 15 18
 5 1 4 7 3 6 2 26 22 25 28 24 27 23 12 8 11 14 10 13 9 19 15 18 21 17 20 16
 3 6 2 5 1 4 7 24 27 23 26 22 25 28 10 13 9 12 8 11 14 17 20 16 19 15 18 21

Nu este asociativ

Este medial

Nu este paramedial

Nu este bicomutativ

Nu este AG grupoid

Nu este GA grupoid

Nu este GA1 grupoid

Nu este AD grupoid

Nu este DA grupoid

Nu este hexagonal

Nu este distributiv la dreapta

Nu este distributiv la stanga

Nu este unitate de dreapta

Nu este unitate de stanga

Nu este unitate

Nu este Ward

Nu este Ward invers

Ex. 3

The screenshot shows a Pascal ABC program window titled "Pascal ABC". The code in the editor is as follows:

```

Program produs_carteziem;
//uses crt;

```

The output window displays a 12x12 multiplication table for the numbers 1 through 12, followed by a list of algebraic properties and their status (Yes/No):

```

1 2 3
2 3 1
3 1 2
1 4 2 3 5 8 6 7 9 12 10 11
3 2 1 1 7 6 5 11 10 12 9
4 1 3 2 5 7 6 12 9 10 11
2 3 1 4 6 7 5 8 10 11 9 12
5 8 5 7 9 12 10 11 1 4 2 3
7 6 3 5 11 10 12 9 5 2 4 1
8 5 7 6 12 9 11 10 4 1 3 2
6 7 5 8 10 11 9 12 2 3 1 4
9 12 10 11 1 4 2 3 5 8 6 7
11 10 12 5 3 2 4 1 7 5 8 9
12 9 11 10 4 1 3 2 5 7 6
11 10 9 12 2 3 1 4 6 7 5 8
Nu este asociativ
Este medial
Nu este paramedial
Nu este bicomutativ
Este AG grupoid
Nu este GA grupoid
Nu este GA1 grupoid
Este AD grupoid
Nu este DA grupoid
Este hexagonal
Este distributiv la dreapta
Este distributiv la stanga
Nu este unitate de dreapta
Nu este unitate de stanga
Nu este unitate
Nu este Ward
Nu este Ward invers

```

Ex.4

```

Pascal ABC
Файл  Правка  Вид  Программа  Сервис  Помощь
[Icons]
*Program2.pas
Program produs_cartezian;
//uscs crt;
type masiv=array[1..100,1..100] of integer;
1 2 4 3
2 3 1 4
4 1 3 2
3 4 2 1
1 2 3 4 5 6 10 11 12 7 8 9
3 1 2 6 4 5 12 10 11 9 7 8
2 3 1 5 6 4 11 12 10 8 9 7
4 5 6 7 8 9 1 2 3 10 11 12
6 4 5 9 7 8 3 1 2 12 10 11
5 6 4 8 9 7 2 3 1 11 12 10
10 11 12 1 2 3 7 8 9 4 5 6
12 10 11 3 1 2 9 7 8 6 4 5
11 12 10 2 3 1 8 9 7 5 6 4
7 8 9 10 11 12 4 5 6 1 2 3
9 7 8 12 10 11 6 4 5 3 1 2
8 9 7 11 12 10 5 6 4 2 3 1
Nu este asociativ
Nu este medial
Nu este paramedial
Este bicomutativ
Nu este AC grupoid
Nu este GA grupoid
Nu este GA1 grupoid
Nu este AD grupoid
Nu este DA grupoid
Nu este hexagonal
Nu este distributiv la dreapta
Nu este distributiv la stanga
Nu este unitate de dreapta
Este unitate de stanga 1
Nu este unitate
Nu este Ward
Nu este Ward invers
Строка: 8  Столбец: 30
  
```

Ex. 5

1473625
 6251473
 4736251
 2514736
 7362514
 5147362
 3625147
 123101112192021789161718456131415
 312121011211920978181617645151314
 231111210202119897171816564141513
 161718456131415123101112192021789

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

Nu este asociativ

Este medial

Nu este paramedial

Nu este bicomutativ

Nu este AG grupoid

Nu este GA grupoid

Nu este GA1 grupoid

Nu este AD grupoid

Nu este DA grupoid

Nu este hexagonal

Nu este distributiv la dreapta

Nu este distributiv la stanga

Nu este unitate de dreapta

Nu este unitate de stanga

Nu este unitate

Nu este Ward

Nu este Ward invers

Ex.7

The screenshot shows the Pascal ABC IDE with a program named 'Program2.pas'. The program prints a 16x16 multiplication table and then checks several algebraic properties of the operation defined in the table.

```

Program produs cartezian;

1 2 4 3
3 4 2 1
2 1 3 4
4 3 1 2
2 1 3 4 6 5 7 8 14 13 15 16 10 9 11 12
3 4 2 1 7 8 6 5 15 16 14 13 11 12 10 9
4 3 1 2 8 7 5 6 16 15 13 14 12 11 9 10
1 2 4 3 5 6 8 7 13 14 16 15 9 10 12 11
10 9 11 12 14 13 15 16 6 5 7 8 2 1 3 4
11 12 10 9 15 16 14 13 7 8 6 5 3 4 2 1
12 11 9 10 16 15 13 14 8 7 5 6 4 3 1 2
9 10 12 11 13 14 16 15 5 6 8 7 1 2 4 3
6 5 7 8 2 1 3 4 10 9 11 12 14 13 15 16
7 8 6 5 3 4 2 1 11 12 10 9 15 16 14 13
8 7 5 6 4 3 1 2 12 11 9 10 16 15 13 14
5 6 8 7 1 2 4 3 9 10 12 11 13 14 16 15
14 13 15 16 10 9 11 12 2 1 3 4 6 5 7 8
15 16 14 13 11 12 10 9 3 4 2 1 7 8 6 5
16 15 13 14 12 11 9 10 4 3 1 2 8 7 5 6
13 14 16 15 9 10 12 11 1 2 4 3 5 6 8 7

Nu este asociativ
Nu este medial
Este paramedial
Nu este bicomutativ
Nu este AG grupoid
Nu este GA grupoid
Nu este GA1 grupoid
Nu este AD grupoid
Nu este DA grupoid
Nu este hexagonal
Nu este distributiv la dreapta
Nu este distributiv la stanga
Nu este unitate de dreapta
Nu este unitate de stanga
Nu este unitate
Nu este Ward
Nu este Ward invers
  
```

Строка: 8 Столбец: 30

Ex.6

The screenshot shows the Pascal ABC IDE with the following content:

```

Pascal ABC
Файл  Правка  Вид  Программа  Сервис  Помощь
[Icons]
•Program2.pas
Program produs cartezian;

4 3 2 1
3 4 1 2
2 1 4 3
1 2 3 4
13 14 15 16 9 10 11 12 5 6 7 8 1 2 3 4
14 15 16 13 10 11 12 9 6 7 8 5 2 3 4 1
15 16 13 14 11 12 9 10 7 8 5 6 3 4 1 2
16 13 14 15 12 9 10 11 8 5 6 7 4 1 2 3
9 10 11 12 13 14 15 16 1 2 3 4 5 6 7 8
10 11 12 9 14 15 16 13 2 3 4 1 6 7 8 5
11 12 9 10 15 16 13 14 3 4 1 2 7 8 5 6
12 9 10 11 16 13 14 15 4 1 2 3 8 5 6 7
5 6 7 8 1 2 3 4 13 14 15 16 9 10 11 12
6 7 8 5 2 3 4 1 14 15 16 13 10 11 12 9
7 8 5 6 3 4 1 2 15 16 13 14 11 12 9 10
8 5 6 7 4 1 2 3 16 13 14 15 12 9 10 11
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
2 3 4 1 6 7 8 5 10 11 12 9 14 15 16 13
3 4 1 2 7 8 5 6 11 12 9 10 15 16 13 14
4 1 2 3 8 5 6 7 12 9 10 11 16 13 14 15
Este asociativ
Este medial
Este paramedial
Este bicomutativ
Este AG grupoid
Este GA grupoid
Este GA1 grupoid
Este AD grupoid
Este DA grupoid
Nu este hexagonal
Nu este distributiv la dreapta
Nu este distributiv la stanga
Nu este unitate de dreapta
Nu este unitate de stanga
Nu este unitate
Nu este Ward
Nu este Ward invers
Строка: 8  Столбец: 30
  
```

Ex.8

Pascal ABC

Файл Правка Вид Программа Сервис Помощь

•Program2.pas

```

Program produs carteziian;

2 3 4 1
3 2 1 4
4 1 2 3
1 4 3 2
8 7 6 5 12 11 10 9 16 15 14 13 4 3 2 1
7 8 5 6 11 12 9 10 15 16 13 14 3 4 1 2
6 5 8 7 10 9 12 11 14 13 16 15 2 1 4 3
5 6 7 8 9 10 11 12 13 14 15 16 1 2 3 4
12 11 10 9 8 7 6 5 4 3 2 1 16 15 14 13
11 12 9 10 7 8 5 6 3 4 1 2 15 16 13 14
10 9 12 11 6 5 8 7 2 1 4 3 14 13 16 15
9 10 11 12 5 6 7 8 1 2 3 4 13 14 15 16
16 15 14 13 4 3 2 1 8 7 6 5 12 11 10 9
15 16 13 14 3 4 1 2 7 8 5 6 11 12 9 10
14 13 16 15 2 1 4 3 6 5 8 7 10 9 12 11
13 14 15 16 1 2 3 4 5 6 7 8 9 10 11 12
4 3 2 1 16 15 14 13 12 11 10 9 8 7 6 5
3 4 1 2 15 16 13 14 11 12 9 10 7 8 5 6
2 1 4 3 14 13 16 15 10 9 12 11 6 5 8 7
1 2 3 4 13 14 15 16 9 10 11 12 5 6 7 8

Este asociativ
Este medial
Este paramedial
Este bicomutativ
Este AG grupoid
Este GA grupoid
Este GA1 grupoid
Este AD grupoid
Este DA grupoid
Nu este hexagonal
Nu este distributiv la dreapta
Nu este distributiv la stanga
Nu este unitate de dreapta
Nu este unitate de stanga
Nu este unitate
Este Ward
Este Ward invers

```

Строка: 8 Столбец: 30

Tabelul proprietăților grupoizilor verificați:

	Ex.1	Ex.2	Ex.3	Ex.4	Ex.5	Ex.6	Ex.7	Ex.8
Asociativ	-	-	-	-	-	+	-	+
Medial	+	+	-	+	+	+	-	+
Paramedial	+	-	-	-	-	+	+	+
Bicomutativ	+	-	+	-	-	+	-	+
AG grupoid	+	-	-	+	-	+	-	+
GA grupoid	-	-	-	-	-	+	-	+
GA1 grupoid	-	-	-	-	-	+	-	+
AD grupoid	-	-	-	+	-	+	-	+
DA grupoid	-	-	-	-	-	+	-	+
Hexagonal	-	-	-	+	-	-	-	-
Distributiv la dreapta	-	-	-	+	-	-	-	-
Distributiv la stanga	-	-	-	+	-	-	-	-
Unitate de dreapta	-	-	-	-	-	-	-	-
Unitate de stanga	1	-	1	-	-	-	-	-
Unitate	-	-	-	-	-	-	-	-
Ward	-	-	-	-	-	-	-	+
Ward invers	+	-	-	-	-	-	-	+

