

# Abstract Algebra with Maple

by *Alec Mihailovs*

## 1 Preface

This manual is intended to be used with *Contemporary Abstract Algebra*, by *J. A. Gallian*, 5th ed., Houghton Mifflin (2002). It was inspired by *Abstract Algebra with GAP* by *J. G. Rainbolt* and *J. A. Gallian*, Houghton Mifflin (2002). The latter manual is available for free downloading from the book's web site and from the authors' websites. I highly recommend, in addition to this Maple manual, solve exercises from both the textbook and the GAP manual.

I thank *Joseph A. Gallian* for his comments and suggestions and my Beautiful and Wonderful Wife, *Bette*, for proofreading.

## 2 An Introduction to Maple

Comparing to other Computer Algebra Systems, such as GAP, Mathematica, Matlab, or MathCAD - Maple is the most user friendly and easy to use. It is available on many platforms, including Windows, Mac, and some UNIXes. The latest information about it can be found on <http://www.maplesoft.com> and <http://www.mapleapps.com>.

### 2.1 Keyboard Shortcuts

Writing this manual, I am using Maple 7 on Windows, so some things, such as keyboard shortcuts, might be different if you are using other operating systems. The best way to get familiar with Maple is to click Help and select New User's Tour (or click Alt+h and n). Supposing that you have walked through that Tour, I will underline just some things we will use often. One thing that might be annoying for some people is the absence of Copy and Paste in the context menus appearing on the screen after right-clicking of the mouse. Instead of it, one has to use either toolbar buttons for copying and pasting, or Ctrl+c for copying and Ctrl+v for pasting. I recommend using Ctrl+c, Ctrl+v as well as Ctrl+x for cutting, Ctrl+z for undoing, Ctrl+a for selecting all, Ctrl+p for printing, Ctrl+s for saving etc. Falling into this habit saves a lot of time.

Useful keyboard shortcuts for typing are Ctrl+t - starts the text mode, Ctrl+r - insert nonexecutable formula at the cursor while you are in the text mode - you have to click Ctrl+t after finishing it to return to the text mode. Ctrl+b, Ctrl+i, and Ctrl+u switch to bold, italics, or underlined scripts. If one is in the Maple Input mode, Shift+Enter allows one to go to the next line without

the executing of the command. Ctrl+k inserts a new execution group before the cursor, and Ctrl+j - after. Maple uses functional keys effectively, too. F3 splits execution groups, F4 joins them, Shift+F3 splits sections, and Shift+F4 joins them. To get help on any command, **plot**, for example, one can either select the word, or just put a cursor somewhere inside it, or in front of it and hit the F1 key. Also, for getting a topic search window, one can type Alt+h, then t. For the full text search, Alt+h, then f.

See other keyboard shortcuts for Windows *here*. In particular, the Windows key allows easy access to many Windows features. Some of my favorites are Win+e - starts Windows Explorer, Win +d - shows the desktop. Surfing the Internet, one of the most useful keys is the Escape key - it stops animated gifs blinking :-) Backspace as well as Alt+(left arrow) returns you to the previous page in the Internet Explorer; Alt+(right arrow) forwards you to the next page (if you returned back before that). Up and down arrows allow one to scroll through the text, both in Maple and in IE.

## 2.2 Colons and Semicolons, % sign

Another thing that one should remember from the very beginning is that one should end any statement either with a semicolon, or a colon. The difference is that Maple shows you the answer after a semicolon and hides it after a colon. For instance,

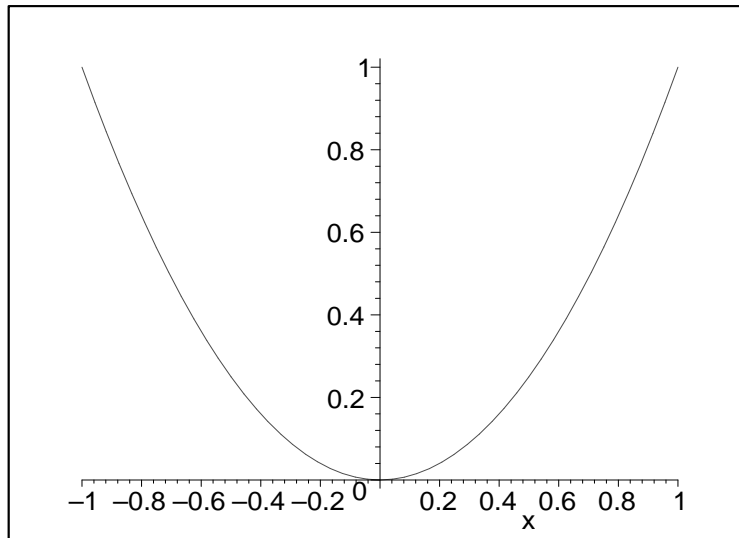
```
> 2^200-1;  
160693804425899027541962092341162602522202993782792835301375
```

prints  $2^{200} - 1$ . If you hit Enter without a colon, or a semicolon, you will get an error. Typing

```
> plot(x^2, x=-1..1):
```

won't produce any visible result, because the command was ended with a colon. To see the picture, we can either change the colon to a semicolon, or type

```
> %;
```



which gives us the plot of the parabola. % in this example means the latest Maple output. If we want to get  $2^{200} - 1$  again, we should use %%%, because it is the 3rd output from the end:

```
> %%%+1;
1606938044258990275541962092341162602522202993782792835301376
```

As we just saw, outputs after a colon were not visible, but had been done by Maple and should be counted to obtain a correct number of the percent signs.

## 2.3 Comments

Generally, the text mode can be used for comments. In the Maple Input mode, # plays the role similar to // in C++, or % in LaTeX - it skips everything after that sign in the line where it is located.

## 2.4 Exercises

1. What is the shortcut Ctrl+f for?
2. Is the find/replace in Maple case sensitive, i.e. gives different results for maple and Maple?
3. How, writing a Maple procedure, one goes to the next line without executing the commands?
- 4-7. Click Ctrl+n, Ctrl+t. In the list of styles on the far left of the context bar, click **Title**. Type "Homework 1" (without quotes). Hit Enter. Change the font in the list of fonts and the font size to anything you like. Type your name and Ctrl+j, Ctrl+. , and click the up-arrow key after that. Type "Exercise 4." (without quotes) and click the down-arrow key. Type 100!; (including a

semicolon) and hit Enter. Type `ifactor(%);` and hit Enter. Click Shift+F3, starting a new exercise, Exercise 5, and add, multiply, subtract and divide a few numbers. Click Shift+F3 again starting Exercise 6 and plot the graph of  $x \sin(\frac{1}{x})$  from  $x = -1$  to  $x = 1$ . Click Shift+F3 again starting Exercise 7. Type `with(plots):` after the Maple prompt and hit Enter. After the Maple prompt, type `animate3d` and click F1. Select the last example on the help page and click Ctrl+c. Click Ctrl+F4. Select `animate3d` and click Ctrl+v. Hit Enter. Click on the picture, then on the play button in the toolbar. Use the mouse to rotate the surface. Find the best-looking position and click the play button again. Click Ctrl+s, choose an appropriate place, give the worksheet a name you like and save it. Click Ctrl+p and print it. Click Alt+f, then e, then h, and save the worksheet as an html file. Click Win+e to start Windows Explorer, find the html file you just saved and click on it to see it in the browser. Enjoy.

## 3 0. Preliminaries

### 3.1 Properties of Integers

It is easy to factor integers in Maple:

```
> ifactor(2^57-1);
(7) (1212847) (524287) (32377)
```

The greatest common divisor and the least common multiple can be evaluated in Maple as follows:

```
> igcd(715,1001);
143
> ilcm(843,216,51);
1031832
```

The next example shows how to find integer solutions of equations:

```
> isolve(7*x+15*y=1);
{y = 1 + 7_Z1, x = -2 - 15_Z1}
```

To find a particular solution, one can replace the unknown `_Z1` by any integer value, for example, by 2:

```
> subs(_Z1=2,%);
{x = -32, y = 15}
```

### 3.2 Modular Arithmetic

Maple can do modular arithmetic, too:

```
> 345 mod 7;
2
```

It can be used for checking the validity of money order numbers, UPS pickup record numbers, ISBN numbers etc.

```
> isMoneyOrder:= n -> n<10^11 and trunc(n/10) mod 9 = n mod 10:
```

This function first checks if the number contains not more than 11 digits and then if the number formed by the first 10 digits is congruent to the last digit, i.e. check digit, modulo 9.

```
> isMoneyOrder(39539881642);
      true
> isMoneyOrder(39559881642);
      false
```

The similar construction works for air ticket numbers and UPS pickup records numbers, just by replacing modulo 9 to modulo 7:

```
> isUPS:= n -> n<10^10 and trunc(n/10) mod 7 = n mod 10:
> isUPS(7681139992);
      true
> isUPS(1213731473673);
      false
> isAirTicket:= n -> n<10^15 and trunc(n/10) mod 7 = n mod 10:
> isAirTicket(1213731473673);
      true
```

For the UPC code, one needs to use a dot product, so we have to load the Linear Algebra package first:

```
> with(LinearAlgebra):
> isUPC:= n -> evalb(Vector(12,convert(n,base,10)) .
> Vector([1,3,1,3,1,3,1,3,1,3,1,3]) mod 10 = 0):
> isUPC(021000658978);
      true
> isUPC(012000658978);
      false
```

A similar construction works for bank checks:

```
> isBankCheck:= n -> evalb(Vector(9,convert(n,base,10)) .
> Vector([9,3,7,9,3,7,9,3,7]) mod 10 = 0):
> isBankCheck(13);
      true
```

For ISBN numbers we need a slightly more sophisticated method, because inputs can include the letter X:

```

> isISBN:= proc(str) local s;
> with(LinearAlgebra);
> if type(str,integer) then s:=[str]; else
> s:=sscanf(str,"%d%[Xx]") end if;
> evalb('if'(nops(s)=1,Vector(10,convert(s[1],base,10)),
> 'if'(nops(s)=2 and not s[2]="", Vector(10,
> [10,op(convert(s[1],base,10))]),Vector([1,0$9]))) . Vector([$
1..10])
> mod 11 = 0) end:
> isISBN(0618122141);
true
> isISBN("618122141");
true
> isISBN("6x");
true

```

As you can see, ISBN numbers without an X can be entered either as integers, or as strings, inside quotes. ISBN numbers with an X at the end must be entered inside quotes, because it is not one of the data formats that Maple recognizes.

### 3.3 Mathematical Induction

Maple can evaluate many sums:

```

> sum(i,i=1..n);

$$\frac{1}{2}(n+1)^2 - \frac{1}{2}n - \frac{1}{2}$$

> simplify(%);

$$\frac{1}{2}n^2 + \frac{1}{2}n$$

> simplify(sum(i^10,i=1..n));

$$\frac{1}{11}n^{11} + \frac{1}{2}n^{10} + \frac{5}{6}n^9 - n^7 - \frac{1}{2}n^3 + n^5 + \frac{5}{66}n$$

> sort(%);

$$\frac{1}{11}n^{11} + \frac{1}{2}n^{10} + \frac{5}{6}n^9 - n^7 + n^5 - \frac{1}{2}n^3 + \frac{5}{66}n$$


```

The answers can be proven by mathematical induction as follows:

```

> f:=n->1/11*n^11+1/2*n^10+5/6*n^9-n^7+n^5-1/2*n^3+5/66*n;
> f(1)=1 and simplify(f(n+1))=simplify(f(n)+(n+1)^10);
true

```

Also, Maple can find formulas for many polynomial sequences, for example, sums of squares, 1, 5, 14, 30, 55, ..., using interpolating polynomial

```
> interp([1..5],[1,5,14,30,55],x);
```

$$\frac{1}{3}x^3 + \frac{1}{2}x^2 + \frac{1}{6}x$$

Since formula is known, Maple can easily continue the sequence:

```
> f:=x->1/3*x^3+1/2*x^2+1/6*x:
> for N from 6 to 10 do f(N) od;
          91
          140
          204
          285
          385
```

### 3.4 Exercises

1. Factor  $3^{100} + 1$ .
2. Find the greatest common divisor and the least common multiple of 670592745, 83810205, 113351790, and 695529645.
3. Find integer solutions of the equation  $42x - 47y = 1$ .
4. Find the last digit of the ISBN number starting from 1-894511-01.
5. Find the formula for the sum of 9th powers of integers from 1 to  $n$ .
6. Find the formula for the elements of the sequence 3, 17, 81, 255, 623, 1293, ... and find the next 4 elements of the sequence.

## 4 1. Introduction to Groups

### 4.1 Cyclic and Dihedral Groups

In this section we will study two series of groups, cyclic and dihedral, represented as rotations and symmetries of regular polygons. It is convenient to enumerate all the vertices of a regular  $n$ -gon counterclockwise from 1 to  $n$ . Now, if a symmetry moves vertex 1 to  $a_1$ , vertex 2 to  $a_2$ , and so on, ..., vertex  $n$  to  $a_n$ , then we can denote it  $[a_1, a_2, \dots, a_n]$ . For example, rotation of  $360/n$  degrees moves vertex 1 to 2, vertex 2 to 3, and so on, ..., vertex  $n$  to 1, so it can be denoted as  $[2, 3, \dots, n, 1]$ , which can be represented in Maple as  $[\$2..n,1]$ . The identity (i.e. no change) will be represented as  $[1, 2, \dots, n]$ , or  $[\$1..n]$  in Maple notation. See Maple help item on "dollar", explaining that notation. Now we can define cyclic and dihedral groups as follows:

```
> cyclic:=n->[seq([$i..n,$1..i-1],i=1..n)]:
> dihedral:=n->[op(cyclic(n)),seq([n+1-j$j=i..n,n+1-j$j=1..i-1],i=1..n)
> ]:
```

Cyclic groups are defined for all positive integers  $n$ , but dihedral groups are defined here only for  $n$  not less than 3:

```
> dihedral(2);
[[1, 2], [2, 1], [2, 1], [1, 2]]
```

Here are correctly defined groups:

```
> cyclic(3);
[[1, 2, 3], [2, 3, 1], [3, 1, 2]]
> dihedral(4);
```

```
[[1, 2, 3, 4], [2, 3, 4, 1], [3, 4, 1, 2], [4, 1, 2, 3], [4, 3, 2, 1], [3, 2, 1, 4], [2, 1, 4, 3],
[1, 4, 3, 2]]
```

In this notation, the last element of  $D_4$  transfers vertex 1 to itself, vertex 2 to the position of vertex 4, vertex 3 to itself, and vertex 4 to the position of vertex 2, so it is the reflection across the diagonal connecting 1 and 3, see the picture below. Permutation notation introduced above, is rather long. To make notation easier, we will denote elements just by their ordinal numbers in the lists of the group elements, so the identity will always be the number 1, and the last element of  $D_4$  will be 8.

In this new notation, to find the inverse elements, we can use the following procedure:

```
> inv:=proc(a,g) local i,v,b,k; k:=nops(g[a]); b:=[0$k]; for i
to k do
> b[g[a][i]]:=i od; member(b,g,'v'); v end:
```

For example,

```
> inv(5,cyclic(12));
9
> inv(5,dihedral(4));
5
```

If we want to see the graphical representations of  $kr$  elements of  $g$ , starting from  $m$ , displaying  $k$  elements in each row, in  $r$  rows, we can use the following procedure:

```

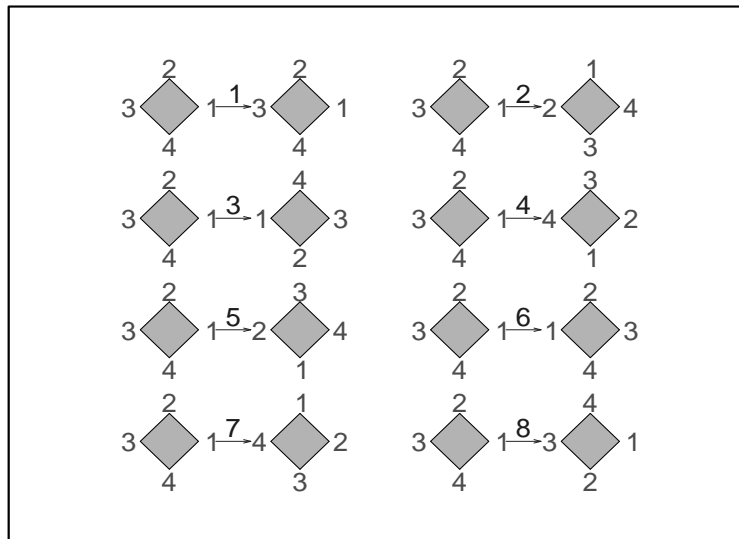
> Grid:=proc(g,m,k,r) local
> ngon,ngons1,ngons2,n,a,b,i,j1,j2,p,ar,l,ngonlabels,text1,text2,txstar;
> with(plots); n:=nops(g[1]);
> ngon := (a,b) -> [seq([ a+cos(2*Pi*i/n), b+sin(2*Pi*i/n) ], i
=
> 1..n)]:
> ngons1:=seq(seq(ngon(10*j1,-4*j2),j1=1..k),j2=1..r);
> ngons2:=seq(seq(ngon(10*j1+4.5,-4*j2),j1=1..k),j2=1..r);
> p:=polygonplot({ngons1,ngons2
> },axes=NONE,scaling=CONSTRAINED,color=aquamarine):
> ar:=arrow([seq(seq([[10*j1+1.6,-4*j2],[1.2,0]],j1=1..k),j2=1..r)],shap
> e=arrow,color=blue):
> ngonlabels:=(a,b,l)->seq([ a+1.4*cos(2*Pi*i/n),
> b+1.4*sin(2*Pi*i/n),l[i+1] ], i = 0..n-1):
> text1:=textplot([seq(seq(ngonlabels(10*j1,-4*j2,[$1..n]),j1=1..k),j2=1
> ..r)],color=red):
> text2:=textplot([seq(seq(ngonlabels(10*j1+4.5,-4*j2,g[inv(m-1+j1+k*(j2
> -1),g)]),j1=1..k),j2=1..r)],color=red):
> txstar:=textplot([seq(seq([10*j1+2.2,-4*j2+.5,m-1+j1+k*(j2-1)],j1=1..k
> ),j2=1..r)],color=blue):
> display(p,ar,text1,text2,txstar) end:

```

Here is the list of elements of  $D_4$ :

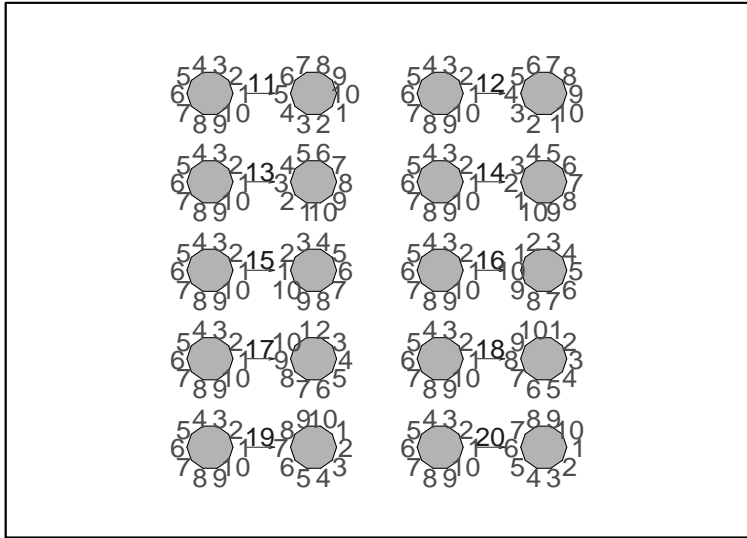
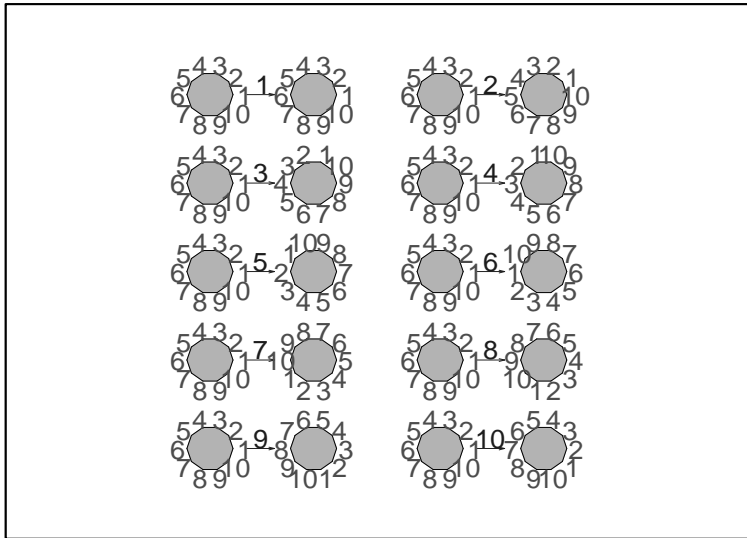
```
> Grid(dihedral(4),1,2,4);
```

Warning, the name `changecoords` has been redefined



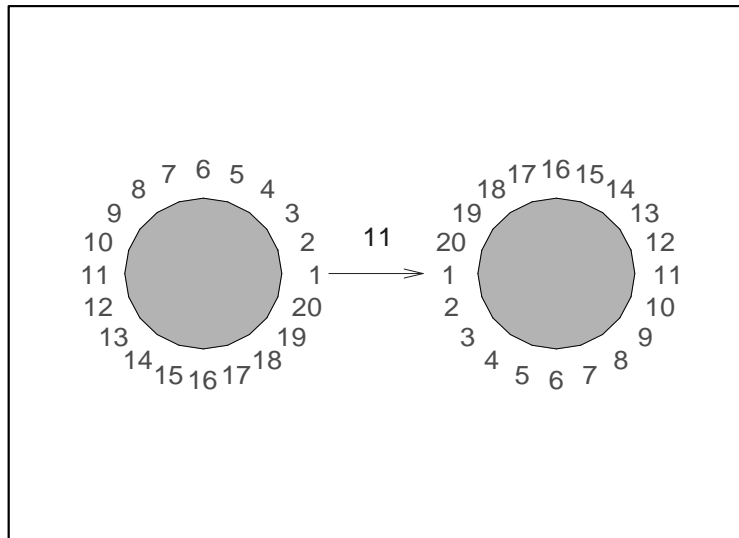
Another example, the list of elements of  $D_{10}$ :

```
> Grid(dihedral(10),1,2,5);Grid(dihedral(10),11,2,5);
```



Also, we can use this procedure to display only one element. For example, 11th element in  $D_{20}$ :

```
> Grid(dihedral(20), 11, 1, 1);
```



If we want to see its representation as a permutation, it can be done as follows:

```
> dihedral(20)[11];
[11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Cyclic group  $C_n$  has  $n$  elements and dihedral group  $D_n$  has  $2n$  elements.

Using the **nops** command can test it:

```
> nops(cyclic(12));
12
> nops(dihedral(100));
200
```

Notice, that all elements of cyclic groups are rotations. The first  $n$  elements of **dihedral**( $n$ ) are rotations, the other  $n$  are reflections.

To multiply elements, we can use the following procedure:

```
> mult:=proc(a,b,g) local i,v;
> member([seq(g[a][g[b][i]],i=1..nops(g[a]))],g,'v');v end;
```

For example,

```
> mult(3,7,cyclic(12));
9
> mult(3,7,dihedral(4));
5
> mult(7,3,dihedral(4));
5
```

## 4.2 Cayley Tables

The following procedure displays the Cayley table of a group:

```
> cayley:=g->Matrix(nops(g),(i,j)->mult(i,j,g)):
```

For example,

```
> cayley(dihedral(4));
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 1 & 8 & 5 & 6 & 7 \\ 3 & 4 & 1 & 2 & 7 & 8 & 5 & 6 \\ 4 & 1 & 2 & 3 & 6 & 7 & 8 & 5 \\ 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \\ 6 & 7 & 8 & 5 & 4 & 1 & 2 & 3 \\ 7 & 8 & 5 & 6 & 3 & 4 & 1 & 2 \\ 8 & 5 & 6 & 7 & 2 & 3 & 4 & 1 \end{bmatrix}$$

It is easy to see that the product of two reflections (i.e. numbers from 5 to 8) is a rotation, and the product of a reflection and a rotation is a rotation.

Another evident thing is that reflections are inverse to themselves. Notice that the matrix is not symmetric because group  $D_4$  is not Abelian. The following procedure is checking whether a group is Abelian:

```
> isAbelian:=g->type(cayley(g),'Matrix'(symmetric)):
```

```
> isAbelian(dihedral(4));
```

*false*

For cyclic groups Cayley tables are very symmetric:

```
> cayley(cyclic(12));
```

[12 x 12 Matrix Data Type : anything Storage : rectangular Order : Fortran\_order]

It is a placeholder. By default, Maple shows them for matrices larger than 10x10. To work with the matrix, one should right-click on the placeholder and use the context menu. In case we want to see the matrix in the worksheet instead of that, we should increase the default size of rtable:

```
> interface(rtablesize=25):
```

```
> %;
```

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

Right-clicking still shows the same context menu.

```
> isAbelian(cyclic(12));
      true
```

### 4.3 Operations

If we need to do a lot of calculations in some specific group,  $D_4$ , for instance, we can define special multiplication and inverse element operations for it:

```
> '&*':=(a,b)->mult(a,b,Group):
```

Before using it, we should specify the group:

```
> Group:=dihedral(4):
> 3&*4;
      2
```

Similarly for the inverse element,

```
> '&-':=a->inv(a,Group):
> &-2;
      4
> 2&*5&*2;
      7
```

One has to be very careful with that though, since the answers depend on the group, and for calculations in other groups we should redefine the **Group**.

```
> Group:=cyclic(12):
> 3&*4;
      6
> 2&*5&*2;
      5
```

## 4.4 Exercises

1. Draw elements of  $D_3$  and  $C_6$ .
2. Draw 15th element of  $D_{24}$ .
3. Represent 15th element of  $D_{24}$  as a permutation.
4. Display Cayley tables of  $D_3$  and  $C_6$ . Are these groups Abelian?
5. Find  $aba^{-1}$  for all pairs of elements  $a$  and  $b$  from  $D_3$  and  $D_4$ .
6. Guess if  $11\pi^{17}27\pi^{-3}$  in  $D_{15}$  is a rotation, or a reflection, and check it out by direct calculation.

## 5 2. Groups

### 5.1 Groups $U(n)$

Here is the definition of groups  $U(n)$  formed by relatively prime with  $n$  integers mod  $n$ . I used the name **un** for them since  $U$  is reserved in Maple for Chebyshev polynomials.

```
> un:=n->select(m->evalb(igcd(m,n)=1),[$1..n]):
> un(12);
[1, 5, 7, 11]
> for N to 100 do T[N]:=nops(un(N)) od:
```

Here are the numbers of elements of groups  $U(n)$  for  $n$  from 1 to 100, written so that 78=24 means that the group  $U(78)$  has 24 elements.

```
> op(op(T));
```

```
[1 = 1, 2 = 1, 3 = 2, 4 = 2, 5 = 4, 6 = 2, 7 = 6, 8 = 4, 9 = 6, 10 = 4, 11 = 10, 12 = 4, 13 = 12,
14 = 6, 15 = 8, 16 = 8, 17 = 16, 18 = 6, 19 = 18, 20 = 8, 21 = 12, 22 = 10, 23 = 22,
24 = 8, 25 = 20, 26 = 12, 27 = 18, 28 = 12, 29 = 28, 30 = 8, 31 = 30, 32 = 16,
33 = 20, 34 = 16, 35 = 24, 36 = 12, 37 = 36, 38 = 18, 39 = 24, 40 = 16, 41 = 40,
42 = 12, 43 = 42, 44 = 20, 45 = 24, 46 = 22, 47 = 46, 48 = 16, 49 = 42, 50 = 20,
51 = 32, 52 = 24, 53 = 52, 54 = 18, 55 = 40, 56 = 24, 57 = 36, 58 = 28, 59 = 58,
60 = 16, 61 = 60, 62 = 30, 63 = 36, 64 = 32, 65 = 48, 66 = 20, 67 = 66, 68 = 32,
69 = 44, 70 = 24, 71 = 70, 72 = 24, 73 = 72, 74 = 36, 75 = 40, 76 = 36, 77 = 60,
78 = 24, 79 = 78, 80 = 32, 81 = 54, 82 = 40, 83 = 82, 84 = 24, 85 = 64, 86 = 42,
87 = 56, 88 = 40, 89 = 88, 90 = 24, 91 = 72, 92 = 44, 93 = 60, 94 = 46, 95 = 72,
96 = 32, 97 = 96, 98 = 42, 99 = 60, 100 = 40]
```

The number theory package has a built in function **phi** for these numbers:

```
> with(numtheory):
```

```
Warning, the protected name order has been redefined and unprotected
```

```
> phi(78);
```

24

Cayley tables for groups  $U(n)$  can be found using the following command:

```
> CayleyU:=n->Matrix(nops(un(n)),(i,j)->un(n)[i]*un(n)[j] mod n):  
> CayleyU(42);
```

$$\begin{bmatrix} 1 & 5 & 11 & 13 & 17 & 19 & 23 & 25 & 29 & 31 & 37 & 41 \\ 5 & 25 & 13 & 23 & 1 & 11 & 31 & 41 & 19 & 29 & 17 & 37 \\ 11 & 13 & 37 & 17 & 19 & 41 & 1 & 23 & 25 & 5 & 29 & 31 \\ 13 & 23 & 17 & 1 & 11 & 37 & 5 & 31 & 41 & 25 & 19 & 29 \\ 17 & 1 & 19 & 11 & 37 & 29 & 13 & 5 & 31 & 23 & 41 & 25 \\ 19 & 11 & 41 & 37 & 29 & 25 & 17 & 13 & 5 & 1 & 31 & 23 \\ 23 & 31 & 1 & 5 & 13 & 17 & 25 & 29 & 37 & 41 & 11 & 19 \\ 25 & 41 & 23 & 31 & 5 & 13 & 29 & 37 & 11 & 19 & 1 & 17 \\ 29 & 19 & 25 & 41 & 31 & 5 & 37 & 11 & 1 & 17 & 23 & 13 \\ 31 & 29 & 5 & 25 & 23 & 1 & 41 & 19 & 17 & 37 & 13 & 11 \\ 37 & 17 & 29 & 19 & 41 & 31 & 11 & 1 & 23 & 13 & 25 & 5 \\ 41 & 37 & 31 & 29 & 25 & 23 & 19 & 17 & 13 & 11 & 5 & 1 \end{bmatrix}$$

To find inverse elements of groups  $U(n)$  one can use the following procedure:

```
> invU:=proc(a,n) local v; igcdex(a,n,'v'); v mod n end:  
> invU(23,42);
```

11

## 5.2 Matrix Groups mod n

Maple is good for calculations with matrix groups. The following examples are self-explanatory.

```
> with(LinearAlgebra):  
> A:=Matrix([[1,2],[3,4]]);
```

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
> B:=Inverse(A) mod 5;
```

$$B := \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}$$

```
> A.B;
```

$$\begin{bmatrix} 11 & 5 \\ 25 & 11 \end{bmatrix}$$

```
> Map(x->x mod 5,%);
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> A^(-1);
```

$$\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & \frac{-1}{2} \end{bmatrix}$$

Maple operates faster with lists than with matrices or Matrices. That's why it is better to define matrix groups using lists instead of Matrices. Groups  $GL(2, Z_n)$  and  $SL(2, Z_n)$  can be defined as follows:

```
> gl2:=n->select(A->evalb(igcd(A[1,1]*A[2,2]-A[1,2]*A[2,1],n)=1),
> [seq(seq(seq(seq([[j,i],[1,k]],l=0..n-1),k=0..n-1),j=0..n-1),i=0..n-1)
> ]):
> sl2:=n->select(A->evalb(A[1,1]*A[2,2]-A[1,2]*A[2,1] mod n=1),
> [seq(seq(seq(seq([[j,i],[1,k]],l=0..n-1),k=0..n-1),j=0..n-1),i=0..n-1)
> ]):
```

Here are the numbers of elements of some of them:

```
> for N from 2 to 6 do nops(sl2(N)) od;
6
24
48
120
144
> for N from 2 to 6 do nops(gl2(N)) od;
6
48
96
480
288
```

Look at some of their elements:

```
> map(matrix,sl2(2));
[[ [1 0], [0 1] ], [ [1 0], [1 1] ], [ [0 1], [1 0] ], [ [0 1], [1 1] ], [ [1 1], [1 0] ], [ [1 1], [0 1] ] ]
> matrix(gl2(6)[256]);
[[ 3 5 ], [ 4 1 ] ]
```

Multiplication in  $GL(n)$  and  $SL(n)$  can be defined as follows:

```
> mm:=(A,B,n)->[[A[1,1]*B[1,1] +A[1,2]*B[2,1] mod n,A[1,1]*B[1,2]
> +A[1,2]*B[2,2] mod n ], [A[2,1]*B[1,1] +A[2,2]*B[2,1] mod
> n,A[2,1]*B[1,2] +A[2,2]*B[2,2] mod n]]:
```

For example,

```
> matrix(mm([[1,2],[3,4]],[[5,6],[7,8]],11));
[[ 8 0 ], [ 10 6 ] ]
```

Inverse elements also can be determined by a direct calculation. In  $SL(n)$  it is especially simple:

```
> invSL:=(A,n)->[[A[2,2],-A[1,2] mod n],[-A[2,1] mod n, A[1,1]]]:
> invGL:=proc(A,n) local d; d:=invU(A[1,1]*A[2,2]-A[1,2]*A[2,1],n);
> [[A[2,2]*d mod n,-A[1,2]*d mod n],[-A[2,1]*d mod n, A[1,1]*d
mod n]]
> end:
```

For example,

```
> matrix(invSL([[3,4],[5,7]],20));
      [ 7 16 ]
      [15  3 ]
> matrix(invGL([[1,2],[3,4]],25));
      [ 23  1 ]
      [14 12 ]
```

### 5.2.1 Note.

We need to include **mm**, **invSL**, and **invGL** inside the **matrix()**, if we want to see the output as a matrix. Otherwise the output would look as a list of matrix rows. To apply the **matrix()** to every element of a set  $S$ , we need to use the command **map(matrix, S)**;

## 5.3 Group Definition

We'll define a group  $G$  in a standard way, as a 2-element list containing a set or a list  $G[1]$  with an associative binary operation  $G[2]$  having an identity and inverses for all elements. To do that, we need to define a few procedures.

The following procedure is checking if the rows of the Cayley table are permutations of the group elements:

```
> isCP:=proc(g,m) local i,j;
> i:=1; while (i<=nops(g) and
> {seq(m(g[i],g[j]),j=1..nops(g))}={op(g)}) do i:=i+1 od;
> evalb(i=nops(g)+1) end:
```

The following procedure is checking associativity:

```
> isAssociative:=proc(g,m) local i,j,k;
> i:=1; j:=1; k:=1; while(i<=nops(g) and
> m(m(g[i],g[j]),g[k])=m(g[i],m(g[j],g[k]))) do if k=nops(g) then
if
> j=nops(g) then i:=i+1; j:=1; k:= 1 else j:=j+1 fi else k:=k+1
fi od;
> evalb(i=nops(g)+1) end:
```

The following example shows that these two procedures are not enough to define a group.

### 5.3.1 Example 1.

Let  $g$  be an arbitrary set containing more than 1 element, and  $m(a,b) = b$  for all pairs  $(a,b)$  of elements of  $g$ . It is not a group, because  $ab = bb$  would imply  $a = b$  in a group. However,  $m$  is associative and every row of the Cayley table is the trivial permutation of the elements of  $g$ :

```
> isCP({0,1},(a,b)->b);
      true
> isAssociative({0,1},(a,b)->b);
      true
```

The following Theorem shows that an additional property of the existence of a right identity is enough for defining a group.

### 5.3.2 Theorem 1.

Let  $m$  be a binary associative operation on a set  $G$  so that

- i*) for every  $g$  in  $G$  the left multiplication by  $g$  is one-to-one and onto  $G$ , i.e. for every element  $f$  of  $G$  there is a unique element  $h$  of  $G$  so that  $f = gh$ ;
- ii*) there exists a right identity  $e$ , so that  $ge=g$  for every element  $g$  of  $G$ .

Then the set  $G$  with the operation  $m$  is a group.

### 5.3.3 Proof.

Pick an element  $g$  in  $G$ . By associativity,  $gh = (ge)h = g(eh)$ , so  $h = eh$  for every  $h$  (by the one-to-one property of the left multiplication by  $g$ ). That means that  $e$  is a left identity as well. By *i*), there exists unique elements  $h$  and  $x$  of  $G$  so that  $gh = e$  and  $hx = e$ . Then,  $x = (gh)x = g(hx) = g$ , in other words,  $gh = hg = e$ , so  $h$  is an inverse element of  $g$ . Since the operation is associative and there exists an identity and inverses of all elements, the set  $G$  with the operation  $m$  is a group.

The following procedure is checking if a set  $g$  with an operation  $m$  (satisfying **isCP**) has a right identity:

```
> hasRightId:=proc(g,m) local i,k;
> member(g[1],[seq(m(g[1],g[i]), i=1..nops(g))], 'k'); i:=1;
> while (i<=nops(g) and m(g[i],g[k])=g[i]) do i:=i+1 od;
> evalb(i=nops(g)+1) end;
```

Here is a negative example:

```
> hasRightId([0,1],(a,b)->b);
      false
```

Finally, we can introduce a new Maple type - a group:

```

> 'type/group':=proc(g)
> type(g,list) and nops(g)=2 and (type(g[1],list) or type(g[1],set))
and
> type(g[2],procedure) and isCP(op(g)) and isAssociative(op(g))
and
> hasRightId(op(g)) end:

```

Let's check a few examples of the groups we know:

```

> type([un(42), (i,j)->i*j mod 42],group);
      true
> type([[1..8], (a,b)->mult(a,b,dihedral(4))],group);
      true
> type([sl2(3), (A,B)->mm(A,B,3)],group);
      true
> type([[0..9], (a,b)->a+b mod 10],group);
      true

```

If we know that the set or list  $g$  with the operation  $m$  is a group, the identity and the inverses can be found as follows:

```

> Id:=proc(g,m) local i,k;
> member(g[1], [seq(m(g[1],g[k]),k=1..nops(g))], 'i');g[i] end:
> Inv:=proc(a,g,m) local i,k;
> member(Id(g,m), [seq(m(a,g[k]),k=1..nops(g))], 'i');g[i] end:

```

For example,

```

> matrix(Id(g12(5), (a,b)->mm(a,b,5)));
      [ 1  0 ]
      [ 0  1 ]
> matrix(Inv([[1,2], [3,4]],g12(5), (a,b)->mm(a,b,5)));
      [ 3  1 ]
      [ 4  2 ]

```

The Cayley table can be displayed as follows:

```

> cayleyTable:=(g,m)->Matrix(nops(g), (i,j)->m(g[i],g[j])):

```

For example, for  $Z_{10}$ :

```

> cayleyTable([0..9], (a,b)->a+b mod 10);
      [ 0  1  2  3  4  5  6  7  8  9 ]
      [ 1  2  3  4  5  6  7  8  9  0 ]
      [ 2  3  4  5  6  7  8  9  0  1 ]
      [ 3  4  5  6  7  8  9  0  1  2 ]
      [ 4  5  6  7  8  9  0  1  2  3 ]
      [ 5  6  7  8  9  0  1  2  3  4 ]
      [ 6  7  8  9  0  1  2  3  4  5 ]
      [ 7  8  9  0  1  2  3  4  5  6 ]
      [ 8  9  0  1  2  3  4  5  6  7 ]
      [ 9  0  1  2  3  4  5  6  7  8 ]

```

We can check if the group is Abelian by checking if the Cayley table is symmetric:

```
> isAbelianGroup:=(g,m)->type(cayleyTable(g,m),'Matrix'(symmetric)):
```

For example,

```
> isAbelianGroup([0..9],(a,b)->a+b mod 10);
```

*true*

```
> isAbelianGroup(sl2(2),(a,b)->mm(a,b,2));
```

*false*

## 5.4 Redefining of Groups

If we knew the group identity and the inverses, we can save time in many calculations. That's why it is convenient to add the identity and the procedure finding the inverse elements to the group definition. We define an extended group as a list  $G$  containing four elements, a set  $G[1]$ , a binary operation (multiplication)  $G[2]$ , the identity  $G[3]$ , and the unary operation (inverse)  $G[4]$ :

```
> 'type/extendedGroup':=proc(g) local i;
> if type(g,list) and nops(g)=4 and type([g[1],g[2]],group) and
> g[2](g[1][1],g[3])=g[1][1]
> then i:=1; while not i=nops(g[1])+1 and
> g[2](g[1][i],g[4](g[1][i]))=g[3] do i:=i+1 od;
> evalb(i=nops(g[1])+1) else false fi end;
```

It might be annoying to enter the same operations repeatedly many times for the groups we know. So we can redefine the groups, including the operations, the identities, and the inverses in their definitions. I'll do that in the order they have appeared in this manual, starting their names with capital letters:

```
> Cyclic:=n->[[1..n],(a,b)->'if'(a+b-1<=n,a+b-1,(a+b-1)-n),1,
> a->'if'(a=1,1,n-a+2)];
> Dihedral:=n->[[1..2*n],(a,b)->mult(a,b,dihedral(n)),1,a->'if'(a=1
> or a>n,a,n-a+2)];
> Un:=n->[un(n),(a,b)->a*b mod n,1,a->invU(a,n)];
> GL2:=n->[gl2(n),(a,b)->mm(a,b,n),[[1,0],[0,1]],a->invGL(a,n)];
> SL2:=n->[sl2(n),(a,b)->mm(a,b,n),[[1,0],[0,1]],a->invSL(a,n)];
> Z:=n->[[0..n-1],(a,b)->a+b mod n,0,a->-a mod n];
```

Now we can test the correctness of the new definitions:

```
> type(Cyclic(10),extendedGroup);
```

*true*

```
> type(Dihedral(5),extendedGroup);
```

*true*

```
> type(Un(12),extendedGroup);
```

*true*

```

> type(GL2(2),extendedGroup);
      true
> type(SL2(3),extendedGroup);
      true
> type(Z(20),extendedGroup);
      true

```

Cayley tables for the extended Groups can be defined as follows:

```

> Cayley:=g->cayleyTable(g[1],g[2]):

```

For example,

```

> Map(matrix,Cayley(SL2(2)));

```

$$\begin{bmatrix} \%1 & \%2 & \%3 & \%4 & \%5 & \%6 \\ \%2 & \%1 & \%4 & \%3 & \%6 & \%5 \\ \%3 & \%5 & \%1 & \%6 & \%2 & \%4 \\ \%4 & \%6 & \%2 & \%5 & \%1 & \%3 \\ \%5 & \%3 & \%6 & \%1 & \%4 & \%2 \\ \%6 & \%4 & \%5 & \%2 & \%3 & \%1 \end{bmatrix}$$

$$\%1 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\%2 := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$\%3 := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\%4 := \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\%5 := \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\%6 := \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The same as we did before, we can check if the group is Abelian by checking if the Cayley table is symmetric:

```

> IsAbelian:=g->isAbelianGroup(g[1],g[2]):

```

For example,

```

> IsAbelian(Z(100));
      true
> IsAbelian(Un(15));
      true
> IsAbelian(GL2(3));
      false

```

Every group  $G$ , defined as a 2-element list, containing a set  $G[1]$  and a binary operation  $G[2]$ , can be easily converted to the extended group by adding the identity and inverse element:

```
> 'convert/extendedGroup':=proc(g) local a, i;
> i:=a->Inv(a,(op(g))); [op(g),Id(op(g)),i] end:
```

For example,

```
> convert([[0..10],(a,b)->a+b mod 11],extendedGroup):
> type(% ,extendedGroup);
true
```

## 5.5 Exercises

1. Find the numbers of elements of groups  $U(5)$ ,  $U(25)$ ,  $U(125)$ , and  $U(625)$ .
2. Display the Cayley table of  $U(40)$ .
3. Find the inverse of 151 in  $U(212)$ .
4. Find the inverse matrix of  $\begin{bmatrix} 137 & 253 \\ 217 & 19 \end{bmatrix} \pmod{321}$ .
5. Find the product of  $\begin{bmatrix} 23 & 45 \\ 56 & 78 \end{bmatrix}$  and  $\begin{bmatrix} 75 & 11 \\ 23 & 51 \end{bmatrix} \pmod{125}$ .
6. Find the numbers of elements of  $GL(2, Z_7)$  and  $SL(2, Z_7)$ .
7. Check if the set  $\{5, 15, 25, 45, 55, 65\}$  is a group under multiplication mod 90. If it is, find the identity element and inverses of every element. Display the Cayley table.

## 6 3. Finite Groups; Subgroups

### 6.1 Orders of Elements

After loading the number theory package (we did it in the previous section), orders of elements of groups  $U(n)$  can be evaluated as follows:

```
> order(7,15);
4
> order(31,42);
6
```

The cyclic subgroup of  $U(n)$  generated by  $a$ , can be found using the following procedure:

```
> cycleU:=(c,n)->[seq(c^(i-1) mod n, i=1..order(c,n))]:
> cycleU(7,15);
[1, 7, 4, 13]
> cycleU(31,42);
[1, 31, 37, 13, 25, 19]
```

In general, the cyclic subgroup generated by an element  $c$  of an (extended) group  $g$  can be found using the following procedure:

```
> Cycle:=proc(c,g) local v,a;
> v:=[g[3]]; a:=c;
> while not a=g[3] do v:=[op(v),a]; a:=g[2](a,c) od; v end;
```

For example, the cyclic subgroup generated by 10 in  $Z_{25}$ :

```
> Cycle(10,Z(25));
[0, 10, 20, 5, 15]
```

Another example,

```
> map(matrix,Cycle([[1,2],[3,4]],GL2(5)));
[[ 1 0 ], [ 1 2 ], [ 2 0 ], [ 2 4 ], [ 4 0 ], [ 4 3 ], [ 3 0 ], [ 3 1 ],
 [ 0 1 ], [ 3 4 ], [ 0 2 ], [ 1 3 ], [ 0 4 ], [ 2 1 ], [ 0 3 ], [ 4 2 ]]
```

The orders of elements can be found as the orders of the cyclic subgroups generated by them:

```
> Ord:=proc(c,g) local a,n;
> a:=c; n:=1;
> while not a=g[3] do n:=n+1; a:=g[2](a,c) od; n end:
> Ord([[1,2],[3,4]],SL2(5));
8
> Ord([[2,3],[4,5]],GL2(11));
60
> Ord(715,Z(1001));
7
```

## 6.2 Center and Centralizers

The centralizer of an element  $a$  of an extended group  $G$  can be determined as follows:

```
> CentralizerE:=proc(a,G) local i,v;
> v:=[]; for i to nops(G[1]) do if G[2](G[1][i],a)=G[2](a,G[1][i])
then
> v:=[op(v),G[1][i]] fi od; v end;
```

For example,

```
> CentralizerE(8,Cyclic(8));
[1, 2, 3, 4, 5, 6, 7, 8]
> CentralizerE(8,Dihedral(4));
[1, 3, 6, 8]
```

The centralizer of a subset  $S$  of an extended group  $G$  can be determined as follows:

```

> CentralizerS:=proc(S,G) local i,j,v;
> v:=[]; for i to nops(G[1]) do j:=1; while not j=nops(S)+1 and
> G[2](G[1][i],S[j])=G[2](S[j],G[1][i]) do j:=j+1 od; if j=nops(S)+1
> then v:=[op(v),G[1][i]] fi od; v end:
> CentralizerS([7,8],Dihedral(4));
[1, 3]

```

The center is the centralizer of the group:

```

> Center:=G->CentralizerS(G[1],G):
> Center(Dihedral(7));
[1]
> Center(Dihedral(10));
[1, 6]
> map(matrix,Center(GL2(3)));
[[ 1 0 ], [ 2 0 ],
 [ 0 1 ], [ 0 2 ]]

```

### 6.2.1 Note.

Why do we need two commands for a centralizer - **CentralizerE** and **CentralizerS**? Why can't we use one command, **Centralizer**, for both cases? The problem is that some elements of a group can be equal to some sets of other elements. For example, a group can contain elements 1, 2, and {1,2}.

What would the **Centralizer**({1,2}) be in that case? The centralizer of the element {1,2}, **CentralizerE**({1,2}), or the centralizer of the subset {1,2}, **CentralizerS**({1,2})? Since we can't distinguish between these two cases by checking the type of an argument, we need two different commands for that.

## 6.3 A Subgroup Test

According to Theorem 3.3 on p. 62 of Dr. Gallian's text, to test whether a finite subset  $H$  is a subgroup of  $G$ , it is enough to check if  $H$  is a subset of  $G$  and it is closed under the group operation of  $G$ . So we can use **isCP** for that:

```

> isSubgroup:=(h,G)->{op(h)} subset {op(G[1])} and isCP(h,G[2]):

```

For example,

```

> isSubgroup(Cycle(8,Z(33)),Z(33));
true
> isSubgroup(Cycle([[1,2],[2,0]],GL2(5)),SL2(5));
true

```

Certainly, cyclic subgroups are subgroups :-) as well as the center and centralizers:

```

> isSubgroup(CentralizerE(11,Dihedral(6)),Dihedral(6));
      true
> isSubgroup(CentralizerS({[[1,2],[3,4]],[[1,3],[2,4]]
> },GL2(5)),GL2(5));
      true
> isSubgroup(Center(SL2(4)),SL2(4));
      true

```

Finally, an opposite example:

```

> isSubgroup(Z(5)[1],Z(10));
      false

```

## 6.4 Exercises

- Find the order of 7 in  $U(100)$  and  $U(11)$ .
- Find the order of 6 in  $Z_7, Z_8, Z_9, Z_{10}, Z_{11}$ , and  $Z_{12}$ .
- Find the cyclic subgroup of  $U(145)$  generated by 19.
- Find the cyclic subgroup of  $GL(2, Z_6)$  generated by  $\begin{bmatrix} 2 & 5 \\ 1 & 2 \end{bmatrix}$ .
- Find the order of cyclic subgroups of  $SL(2, Z_{11})$  and  $SL(2, Z_{12})$  generated by  $\begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix}$ .
- Find the centralizer of 3 in the dihedral group  $D_6$ .
- Find the centralizer of  $\left\{ \begin{bmatrix} 1 & 3 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right\}$  in  $GL(2, Z_5)$ .
- Find the center of  $SL(2, Z_6)$ .
- Test if the set  $\{1, 4, 11, 14, 16, 19, 26, 29, 31, 34, 41, 44\}$  is a subgroup of  $U(45)$ .

## 7 4. Cyclic Groups

### 7.1 Primitive Roots

Group  $Z_n$  has  $\phi(n)$  generators. To find them, we can use procedure **un**. For example,

```

> un(20);
      [1, 3, 7, 9, 11, 13, 17, 19]

```

The generators of  $U(n)$  if they exist, are called *primitive roots mod n*. One of them can be found using the function **primroot** from the **numtheory** package that we already loaded in section 2. For example,

```

> primroot(43);

```

It fails when the group  $U(n)$  is not cyclic, so it doesn't have a generator. For example,

```
> primroot(45);
```

*FAIL*

To find all generators, we can use the following procedure:

```
> primroots:=n->{seq(primroot(n)^un(phi(n))[i] mod n,
> i=1..phi(phi(n)))}:

```

For example,

```
> primroots(43);
```

{3, 5, 12, 18, 19, 20, 26, 28, 29, 30, 33, 34}

```
> primroots(45);
```

{*FAIL*}

The number of generators of  $U(n)$  equals  $\phi(\phi(n))$  when it is a cyclic group. For example,

```
> phi(phi(43));
```

12

## 7.2 Elements of Order $d$

Theorem 4.4 on p. 80 of Dr. Gallian's text tells us that if  $d$  is a positive divisor of  $n$ , then there are  $\phi(d)$  elements of order  $d$  in  $Z_n$ . The following procedure lists all of them:

```
> nordlist:=(d,n)->'if'(type(n/d,integer),map(x->x*n/d mod
> n,un(d)),[]):

```

For example, the list of elements of order 20 in  $Z_{100}$ :

```
> nordlist(20,100);
```

[5, 15, 35, 45, 55, 65, 85, 95]

The following procedure counts the number of elements of order  $d$  in  $U(n)$ :

```
> nordU:=proc(d,n) local i, k;
> k:=0; for i from 1 to phi(n) do if order(un(n)[i],n)=d then k:=k+1
fi
> od; k end:

```

For example, the number of elements of order 2 in  $U(45)$ :

```
> nordU(2,45);
```

3

It immediately implies that  $U(45)$  is not a cyclic group, because a cyclic group might have either 0 elements of order 2 if it has an odd order, or 1 element of order 2 if it has an even order.

The following procedure counts the number of elements of order  $d$  in an extended group  $G$ :

```
> Nord:=proc(d,g) local i, k;
> k:=0; for i from 1 to nops(g[1]) do if Ord(g[1][i],g)=d then
k:=k+1 fi
> od; k end;
```

For example, the number of elements of order 10 in  $SL(2, Z_5)$ :

```
> Nord(10,SL2(5));
24
```

Notice that

```
> phi(10);
4
```

and 24 is divisible by  $\phi(10) = 4$ , as it is supposed to be according to the Corollary on p. 80 of Dr. Gallian's book.

Another example, the number of elements of order 2 in groups  $GL(2, Z_n)$  for  $n$  from 2 to 6:

```
> for n from 2 to 6 do Nord(2,GL2(n)) od;
3
13
27
31
55
```

Try to find this sequence in *Neil Sloane's On-Line Encyclopedia of Integer Sequences*. Certainly, for every specific group, one can write a program calculating the number of elements of given order faster. For instance, for the elements of order 2 in  $GL(2, Z_n)$ ,

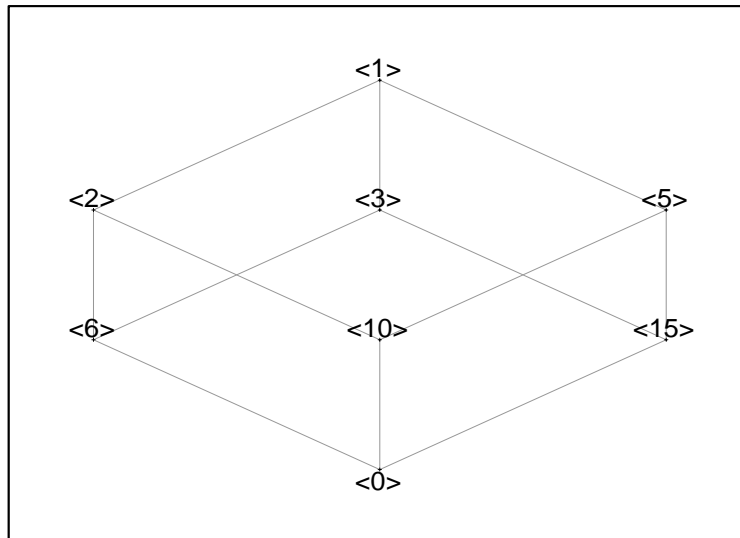
```
> ord2inGL2:=proc(n) local a,b,c,d,N; N:=0;
> for a from 0 to n-1 do for b from 0 to n-1 do for c from 0 to
n-1 do
> for d from 0 to n-1 do
> if a^2+b*c mod n = 1 and b*(a+d) mod n = 0 and c*(a+d) mod n
= 0 and
> d^2+b*c mod n =1
> then N:=N+1 fi od od od od; N-1 end;
```

The following procedure lists the elements of order  $d$  in an extended group  $G$ :

```
> Nordlist:=proc(d,g) local i, v; v:=[];
> for i from 1 to nops(g[1]) do if Ord(g[1][i],g)=d then
> v:=[op(v),g[1][i]] fi od; v end;
```

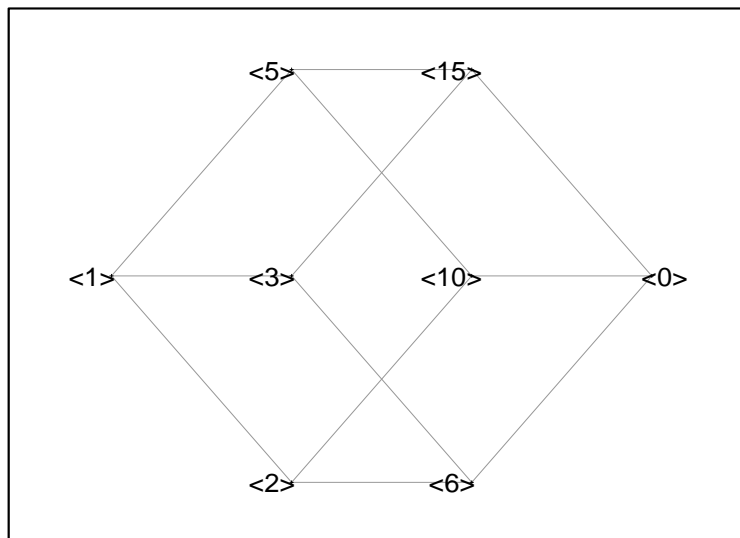
For example, here is the list of elements of order 2 in  $GL(2, Z_5)$ :





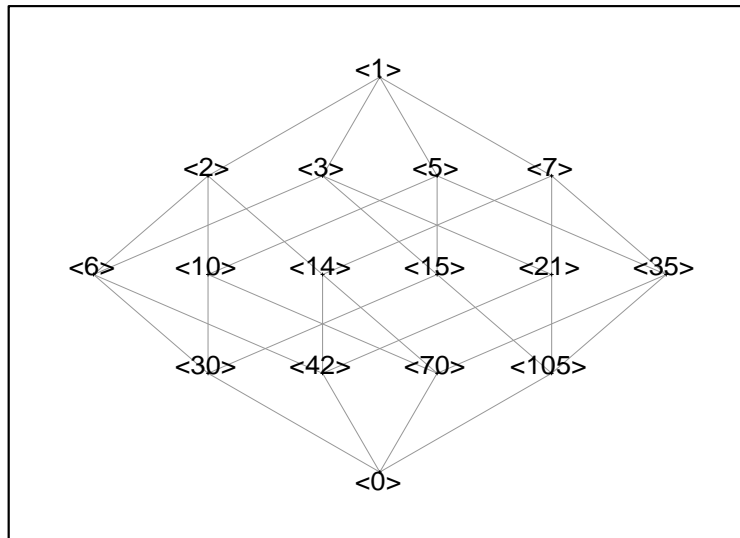
It looks like a cube, doesn't it? Even more after rotating the picture by 90 degrees:

```
> rotate(%,Pi/2);
```



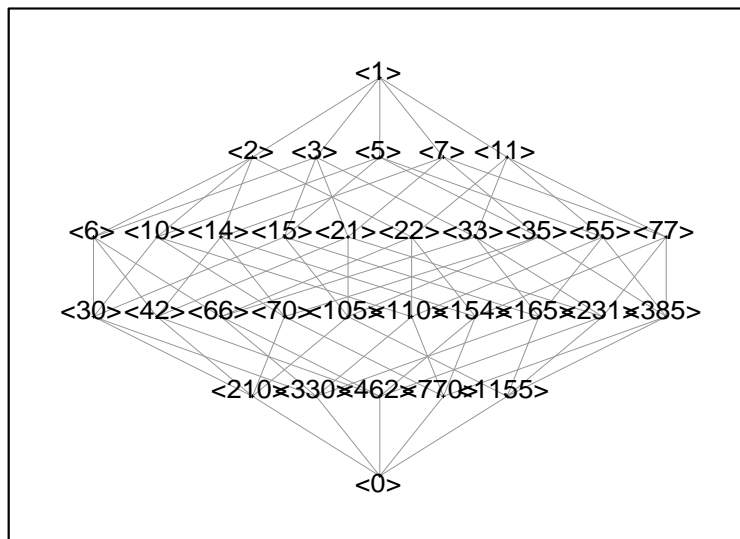
Another example, for  $n = 210$  (a 4-dimensional hypercube:-)

```
> subZ(210);
```



Don't stop at that, draw a 5-dimensional hypercube!

```
> subZ(2310);
```



Number of subgroups of  $Z_n$  equals the number of divisors of  $n$ , which can be evaluated using the function **tau**:

```
> tau(30);
```

8

```
> tau(210);
```

16

```
> tau(2310);
```

32

## 7.4 Exercises.

1. Find which of the groups  $U(n)$  with  $n$  from 46 to 54 are cyclic, and find the generators for them.

2. Find the number of elements of the order 10 in  $Z_{20}$  and list all of them.

3. Find the number of elements of the order 2 in  $U(24)$ .

4. Find the number of elements of the order 12 in  $SL(2, Z_6)$ .

5. Find the number of elements of the order 2 in  $GL(2, Z_n)$  for  $n$  from 7 to 20.

6. List the elements of the order 3 in  $SL(2, Z_3)$ .

7. Draw subgroup lattices of  $Z_8, Z_{12}, Z_{60}$  and  $Z_{100}$ .

## 8 Supplement for Chapters 1 - 4

### 8.1 Subgroups Generated by a Few Elements

We already have the procedure `cycleU` listing the elements of a cyclic subgroup of  $U(n)$ . Here is the procedure finding elements of the subgroup of  $U(n)$ , generated by a set  $s$  of elements of  $U(n)$ :

```
> genU:=proc(s,n) local i,j,v,vs;
> v:={1}; vs:={op(s)} union {1};
> while not v=vs do v:=vs; for i from 1 to nops(v) do for j from
1 to
> nops(s) do
> vs:=vs union {v[i]*s[j] mod n} od od od; v end;
```

For example, here is the subgroup of  $U(48)$  generated by 5 and 7:

```
> genU({5,7},48);
{1, 5, 7, 11, 25, 29, 31, 35}
> un(48);
[1, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47]
```

For a comparison, here are the cyclic subgroups generated by 5 and 7:

```
> cycleU(5,48);
[1, 5, 25, 29]
> cycleU(7,48);
[1, 7]
```

Here is the analogous procedure for extended groups:

```

> Gen:=proc(s,g) local i,j,v,vs;
> v:={g[3]}; vs:={op(s)} union v;
> while not v=vs do v:=vs; for i from 1 to nops(v) do for j from
1 to
> nops(s) do
> vs:=vs union {g[2](v[i],s[j]), g[2](s[j],v[i])} od od od; v end:

```

For  $Z_n$ , we don't actually need such a procedure, because the subgroup generated by  $s = \{a, b, \dots, c\}$  is just a cyclic subgroup generated by  $\mathbf{igcd}(\mathbf{op}(s))$ . Let's see what we get using **Gen**:

```

> Gen({4,6},Z(12));
{0, 2, 4, 6, 8, 10}

```

The same answer! Now, something a little bit more complicated, a subgroup of

$SL(2, Z_3)$  generated by  $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$  and  $\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$ :

```

> map(matrix,Gen({[[1,1],[0,1]],[[0,1],[2,0]]},GL2(3)));

```

$$\left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}, \right.$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix},$$

$$\left. \begin{bmatrix} 2 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} \right\}$$

It looks like  $SL(2, Z_3)$ . Let's check it out:

```

> evalb(Gen({[[1,1],[0,1]],[[0,1],[2,0]]},GL2(3))={op(s12(3))});
true

```

## 8.2 Intersections and Products of Groups

Some exercises in Dr. Gallian's text are using intersections of subgroups and a product of groups. The intersections can be found using the Maple command **intersect**. For example, the intersection of subgroups of  $U(48)$  generated by  $\{5,7\}$  and  $\{9,25\}$ :

```

> genU({5,7},48) intersect genU({9,25},48);
{1, 25}

```

Another example, the intersection of cyclic subgroups of  $Z_{120}$  generated by 4, 6, and 25. :

```

> 'intersect'({op(Cycle(4, Z(120)))}, {op(Cycle(6, Z(120)))},
> {op(Cycle(25, Z(120))));
{0, 60}

```

It must be the cyclic subgroup of  $Z_{120}$  generated by the least common multiple of 4, 6, and 25 mod 120:

```
> Cycle(ilcm(4,6,25) mod 120, Z(120));
      [0, 60]
```

Now, define the product of the extended groups using the following procedure:

```
> ' &x ' := proc(g,h) local m,i;
> m:=(a,b)->[g[2](a[1],b[1]),h[2](a[2],b[2])];
> i:=a->[g[4](a[1]),h[4](a[2])];
> [[seq(seq([g[1][i],h[1][j]],j=1..nops(h[1])),i=1..nops(g[1]))],
> m, [g[3],h[3]], i] end;
```

For example, the product of  $Z_5$  and  $Z_7$ :

```
> A:=Z(5) &x Z(7);
```

Check if it is an extended group:

```
> type(A, extendedGroup);
      true
```

Find the order of it:

```
> nops(A[1]);
      35
```

We can use it to find products of three or more groups, too:

```
> B:=Z(2) &x Z(2) &x GL2(2);
> type(B, extendedGroup);
      true

> nops(B[1]);
      24
```

Here is the identity of it:

```
> B[3];
      [[0, 0], [[1, 0], [0, 1]]]
```

It looks as if  $B$  equals the product of the first two groups multiplied by the third group. Check if it is true:

```
> evalb( B[1] = ((Z(2) &x Z(2)) &x GL2(2))[1] );
      true
```

### 8.3 Is a Group Cyclic?

The following procedure is checking whether an extended group  $G$  is cyclic:

```
> IsCyclic:=proc(g) local v;
> v:={op(g[1])}; while not nops(v)=0 and not Ord(v[1],g)=nops(g[1])
do
> v:=v minus {op(Cycle(v[1],g))} od; not evalb(v={}) end;
```

Here are a few examples:

```

> IsCyclic(Z(1));
true
> IsCyclic(Cyclic(25));
true
> IsCyclic(Dihedral(12));
false
> IsCyclic(Un(48));
false
> IsCyclic(Un(49));
true
> IsCyclic(Z(4) &x Z(6));
false
> IsCyclic(Z(4) &x Z(5));
true

```

If a group is cyclic, the following procedure finds its generators:

```

> Generators:=proc(g) local v,n; n:=nops(g[1]); if n=1 then g[3]
else
> v:={op(g[1])}; while not nops(v)=0 and not Ord(v[1],g)=n do v:=v
> minus {op(Cycle(v[1],g))} od; if v={} then FAIL else
> {seq(Cycle(v[1],g)[un(n)[i]+1],i=1..phi(n))} fi fi end:

```

For example,

```

> Generators(Un(49));
{3, 5, 10, 12, 17, 24, 26, 33, 38, 40, 45, 47}
> Generators(Z(4) &x Z(5));
{[1, 3], [3, 3], [3, 2], [3, 1], [3, 4], [1, 4], [1, 2], [1, 1]}
> Generators(Un(48));
FAIL

```

## 8.4 Normalizers and Conjugacy Classes

"Normalizer" is an environment variable in Maple, so we should use another term for the normalizer of a subgroup. I chose to use "normalizer" even if it contradicts the agreement of using capital letters for commands related to the extended groups. First we have to define conjugate groups:

```

> Conjugate:=(x,h,g)->map(y->g[2](g[2](x,y),g[4](x)),h):

```

For example, the group  $x H x^{-1}$  conjugate to the subgroup  $H$  of  $D_3$  generated by the 6th element, for  $x = 2$ , is

```

> Conjugate(2,[1,6],Dihedral(3));
[1, 4]

```

Now we define normalizers by the following procedure:

```
> normalizer:=(h,g)->select(x->evalb({op(Conjugate(x,h,g))}={op(h)}),g[1]):
```

For example, the normalizer of the cyclic subgroup of  $GL(2, Z_3)$  generated by

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} :$$

```
> map(matrix,normalizer(Cycle([[1,1],[0,1]],GL2(3)),GL2(3)));
```

$$\left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 0 & 2 \end{bmatrix} \right]$$

Conjugacy classes can be defined as follows (also without the capitalization):

```
> c1:=(a,g)->{seq(g[2](g[2](g[1][i]),a),g[4](g[1][i])),i=1..nops(g[1])}
> }:
```

For example, the conjugacy class of the 6th element of  $D_3$ :

```
> c1(6,Dihedral(3));
```

$$\{4, 5, 6\}$$

Another example, the conjugacy class of  $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$  in  $GL(2, Z_3)$ :

```
> map(matrix,c1([[1,1],[0,1]],GL2(3)));
```

$$\left\{ \begin{bmatrix} 2 & 2 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \right\}$$

## 8.5 Group of Quaternions

All the groups of order less than 12 are either groups in our list of extended groups, or their products, except the group of quaternions, see *Finite Groups* article at MathWorld, for example. We will add the group of quaternions  $Q_8$  to our list. It is a non-Abelian group of order 8 having 1 element of order 1 (the identity), 1 element of order 2 (negative 1), and 6 elements of order 4 (plus or minus  $i, j, k$ ). There are only 2 non-Abelian groups of order 8,  $Q_8$  and  $D_4$ . The dihedral group  $D_4$  has only 2 elements of order 4, the rotations by 90 degrees and 270 degrees. Its other elements have degrees either 1 (identity), or 2 (the reflections and the rotation by 180 degrees). Thus, if we find a non-Abelian group of order 8 having more than 2 elements of order 4, it will be the group of quaternions. Try to find it among the subgroups of the groups we already know.  $GL(2, Z_2)$  is too small to contain  $Q_8$  as a subgroup, it has only 6 elements. Try the next smallest groups in the  $GL$  and  $SL$  series,  $SL(2, Z_3)$ . First, find its elements of order 4:

```

> Q:=Nordlist(4,SL2(3)):
> map(matrix,Q);
[[ 0 1 ], [ 1 1 ], [ 2 1 ], [ 0 2 ], [ 1 2 ], [ 2 2 ],
 [ 2 0 ], [ 1 2 ], [ 1 1 ], [ 1 0 ], [ 2 2 ], [ 2 1 ]]

```

Exactly 6 elements. If the group generated by them has 8 elements, then it is the group of quaternions.

```

> map(matrix,Gen(Q,SL2(3)));
{ [[ 2 1 ], [ 1 1 ], [ 0 1 ], [ 1 2 ], [ 0 2 ], [ 2 2 ], [ 1 0 ], [ 2 0 ],
  [ 1 1 ], [ 1 2 ], [ 2 0 ], [ 2 2 ], [ 1 0 ], [ 2 1 ], [ 0 1 ], [ 0 2 ] }

```

8 elements, so it is the group of quaternions. Let's make an extended group from it:

```

> Q8inSL:=[[[1,0],[0,1]],[[2,0],[0,2]],op(Q)],(a,b)->mm(a,b,3),[[1,0],
> [0,1]],a->invSL(a,3)];

```

We also need a function converting the matrices, or lists, to the standard  $i, j, k$  expressions. Let's do it:

```

> quat([[1,0],[0,1]]):=1: quat([[2,0],[0,2]]):=-1: quat(Q[1]):=i:
> quat(Q[2]):=j:
> quat(Q[3]):=-k: quat(Q[4]):=-i: quat(Q[5]):=k: quat(Q[6]):=-j:

```

We will also need the backward conversion, from  $i, j, k$  to the lists:

```

> qback(1):=[[1,0],[0,1]]: qback(-1):=[[2,0],[0,2]]: qback(i):=Q[1]:
> qback(j):=Q[2]:
> qback(-k):=Q[3]: qback(-i):=Q[4]: qback(k):=Q[5]: qback(-j):=Q[6]:

```

Now we can redefine the quaternion group in terms of  $i, j$ , and  $k$ :

```

> Q8:= [map(quat,Q8inSL[1]),(a,b)->quat(Q8inSL[2](qback(a),qback(b))),1,
> a->quat(Q8inSL[4](qback(a)))] :

```

For example,

```

> Center(Q8);
[1, -1]

> c1(i,Q8);
{i, -i}

> Cycle(i,Q8);
[1, i, -1, -i]

```

Certainly, we could define the group of quaternions directly from its Cayley table on p. 89 of Dr. Gallian's text.

## 8.6 Exercises

- Find the subgroup of  $U(96)$  generated by 5 and 7.
- Find the subgroup of  $GL(2, Z_4)$  generated by  $\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$  and  $\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$ .

## 9 Selected Answers

### 9.1 An Introduction to Maple

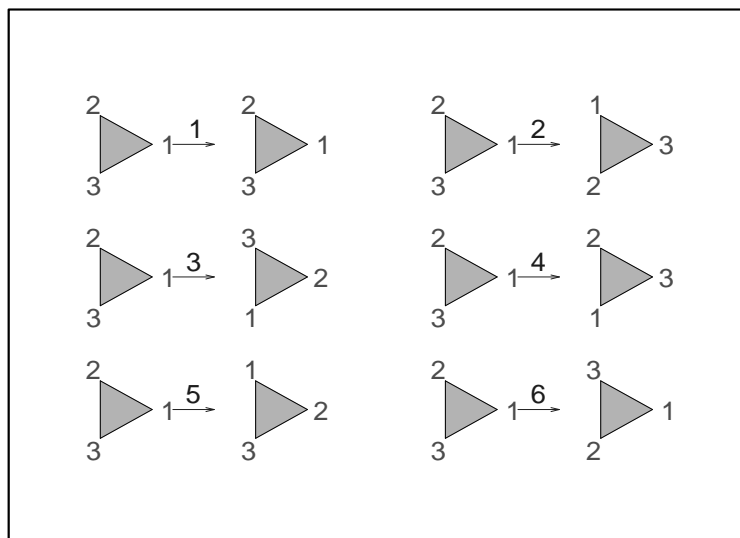
- Find/Replace.
- Yes.
- Shift+Enter.

### 9.2 0. Preliminaries

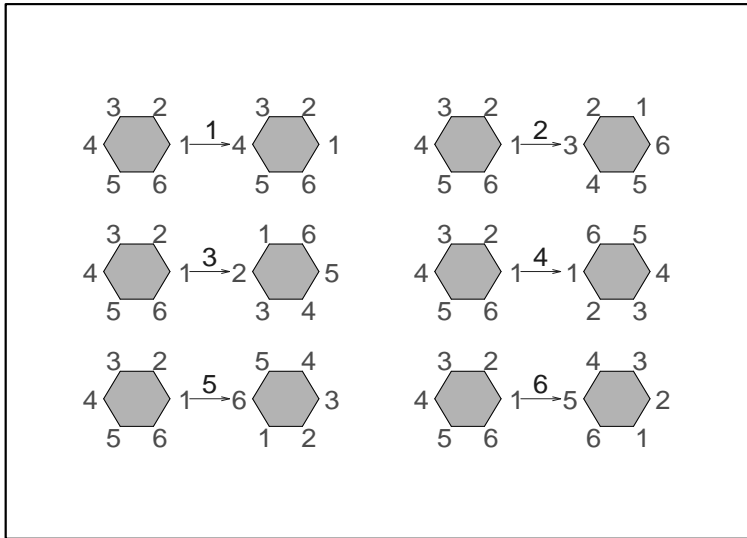
- (2) (41) (61459926512826500975801) (18055139801) (42521761) (133201).
- 12345 and 785064327644945684970.
- $\{x = 28 + 47 \cdot Z1, y = 25 + 42 \cdot Z1\}$ .
- 8.
- $\frac{1n^{10}}{10} + \frac{1n^9}{2} + \frac{3n^8}{4} - \frac{7n^6}{10} + \frac{1n^4}{2} - \frac{3n^2}{20}$ .
- $n^4 - n + 3$  and 2397, 4091, 6555, 9993.

### 9.3 1. Introduction to Groups

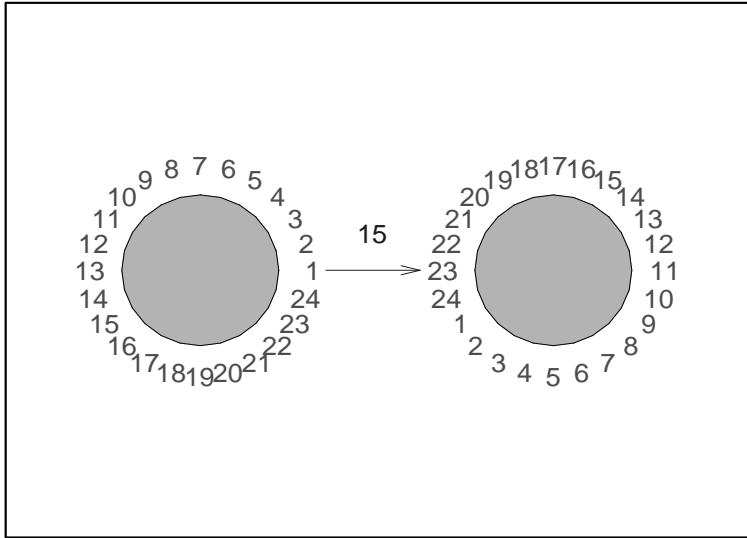
- The elements of  $D_3$ :



The elements of  $C_6$ :



2. The 15th element of  $D_{24}$ :



3. [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

4. The Cayley table of  $D_3$ :

1	2	3	4	5	6
2	3	1	6	4	5
3	1	2	5	6	4
4	5	6	1	2	3
5	6	4	3	1	2
6	4	5	2	3	1

The Cayley table of  $C_3$ :

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 4 & 5 & 6 & 1 \\ 3 & 4 & 5 & 6 & 1 & 2 \\ 4 & 5 & 6 & 1 & 2 & 3 \\ 5 & 6 & 1 & 2 & 3 & 4 \\ 6 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

$C_3$

is Abelian,  $D_3$  is not.

5. The  $aba^{-1}$ -table for  $D_3$ :

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 2 & 3 & 5 & 6 & 4 \\ 1 & 2 & 3 & 6 & 4 & 5 \\ 1 & 3 & 2 & 4 & 6 & 5 \\ 1 & 3 & 2 & 6 & 5 & 4 \\ 1 & 3 & 2 & 5 & 4 & 6 \end{bmatrix}$$

The  $aba^{-1}$ -table for  $D_4$ :

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 7 & 8 & 5 & 6 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 7 & 8 & 5 & 6 \\ 1 & 4 & 3 & 2 & 5 & 8 & 7 & 6 \\ 1 & 4 & 3 & 2 & 7 & 6 & 5 & 8 \\ 1 & 4 & 3 & 2 & 5 & 8 & 7 & 6 \\ 1 & 4 & 3 & 2 & 7 & 6 & 5 & 8 \end{bmatrix}$$

6. Rotation, 4.

## 9.4 2. Groups

1. 4, 20, 100, 500.

2. The Cayley table of  $U(40)$ :

$$\begin{bmatrix} 1 & 3 & 7 & 9 & 11 & 13 & 17 & 19 & 21 & 23 & 27 & 29 & 31 & 33 & 37 & 39 \\ 3 & 9 & 21 & 27 & 33 & 39 & 11 & 17 & 23 & 29 & 1 & 7 & 13 & 19 & 31 & 37 \\ 7 & 21 & 9 & 23 & 37 & 11 & 39 & 13 & 27 & 1 & 29 & 3 & 17 & 31 & 19 & 33 \\ 9 & 27 & 23 & 1 & 19 & 37 & 33 & 11 & 29 & 7 & 3 & 21 & 39 & 17 & 13 & 31 \\ 11 & 33 & 37 & 19 & 1 & 23 & 27 & 9 & 31 & 13 & 17 & 39 & 21 & 3 & 7 & 29 \\ 13 & 39 & 11 & 37 & 23 & 9 & 21 & 7 & 33 & 19 & 31 & 17 & 3 & 29 & 1 & 27 \\ 17 & 11 & 39 & 33 & 27 & 21 & 9 & 3 & 37 & 31 & 19 & 13 & 7 & 1 & 29 & 23 \\ 19 & 17 & 13 & 11 & 9 & 7 & 3 & 1 & 39 & 37 & 33 & 31 & 29 & 27 & 23 & 21 \\ 21 & 23 & 27 & 29 & 31 & 33 & 37 & 39 & 1 & 3 & 7 & 9 & 11 & 13 & 17 & 19 \\ 23 & 29 & 1 & 7 & 13 & 19 & 31 & 37 & 3 & 9 & 21 & 27 & 33 & 39 & 11 & 17 \\ 27 & 1 & 29 & 3 & 17 & 31 & 19 & 33 & 7 & 21 & 9 & 23 & 37 & 11 & 39 & 13 \\ 29 & 7 & 3 & 21 & 39 & 17 & 13 & 31 & 9 & 27 & 23 & 1 & 19 & 37 & 33 & 11 \\ 31 & 13 & 17 & 39 & 21 & 3 & 7 & 29 & 11 & 33 & 37 & 19 & 1 & 23 & 27 & 9 \\ 33 & 19 & 31 & 17 & 3 & 29 & 1 & 27 & 13 & 39 & 11 & 37 & 23 & 9 & 21 & 7 \\ 37 & 31 & 19 & 13 & 7 & 1 & 29 & 23 & 17 & 11 & 39 & 33 & 27 & 21 & 9 & 3 \\ 39 & 37 & 33 & 31 & 29 & 27 & 23 & 21 & 19 & 17 & 13 & 11 & 9 & 7 & 3 & 1 \end{bmatrix}$$

3. 139.

4.  $\begin{bmatrix} 142 & 221 \\ 17 & 44 \end{bmatrix}$ .

5.  $\begin{bmatrix} 10 & 48 \\ 119 & 94 \end{bmatrix}$ .

6. 2016 and 336.

7. It is a group. The identity is 15, the inverses to 5, 15, 25, 45, 55, 65 are 45, 15, 65, 5, 55, 25, in the same order. The Cayley table of it is

$$\begin{bmatrix} 25 & 5 & 55 & 15 & 65 & 45 \\ 5 & 15 & 25 & 45 & 55 & 65 \\ 55 & 25 & 65 & 5 & 45 & 15 \\ 15 & 45 & 5 & 65 & 25 & 55 \\ 65 & 55 & 45 & 25 & 15 & 5 \\ 45 & 65 & 15 & 55 & 5 & 25 \end{bmatrix}$$

### 9.5 3. Finite Groups; Subgroups

1. 4 and 10.

2. 7435112.

3. [1, 19, 71, 44, 111, 79, 51, 99, 141, 69, 6, 114, 136, 119, 86, 39, 16, 14, 121, 124, 36, 104, 91, 134, 81, 89,

4.  $\left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 5 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 2 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 5 & 2 \end{bmatrix}, \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \begin{bmatrix} 4 & 1 \\ 5 & 4 \end{bmatrix}, \begin{bmatrix} 3 & 4 \\ 2 & 3 \end{bmatrix}, \begin{bmatrix} 4 & 5 \\ 1 & 4 \end{bmatrix} \right]$

5. 10 and 12.

6. [1, 2, 3, 4, 5, 6].

7.  $\left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \right]$ .

8.  $\left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \right]$ .

9. Yes.

### 9.6 4. Cyclic Groups

1.  $U(46)$ ,  $U(47)$ ,  $U(49)$ ,  $U(50)$ ,  $U(53)$ , and  $U(54)$  are cyclic. Their generators are

$$\{5, 7, 11, 15, 17, 19, 21, 33, 37, 43\}$$

for  $U(46)$ ,

$$\{5, 10, 11, 13, 15, 19, 20, 22, 23, 26, 29, 30, 31, 33, 35, 38, 39, 40, 41, 43, 44, 45\}$$

for  $U(47)$ ,

$$\{3, 5, 10, 12, 17, 24, 26, 33, 38, 40, 45, 47\}$$

for  $U(49)$ ,

$$\{3, 13, 17, 23, 27, 33, 37, 47\}$$

for  $U(50)$ ,

$$\{2, 3, 5, 8, 12, 14, 18, 19, 20, 21, 22, 26, 27, 31, 32, 33, 34, 35, 39, 41, 45, 48, 50, 51\}$$

for  $U(53)$ ,

$$\{5, 11, 23, 29, 41, 47\}$$

for  $U(54)$ .

2. 4 elements; 2, 6, 14, 18.

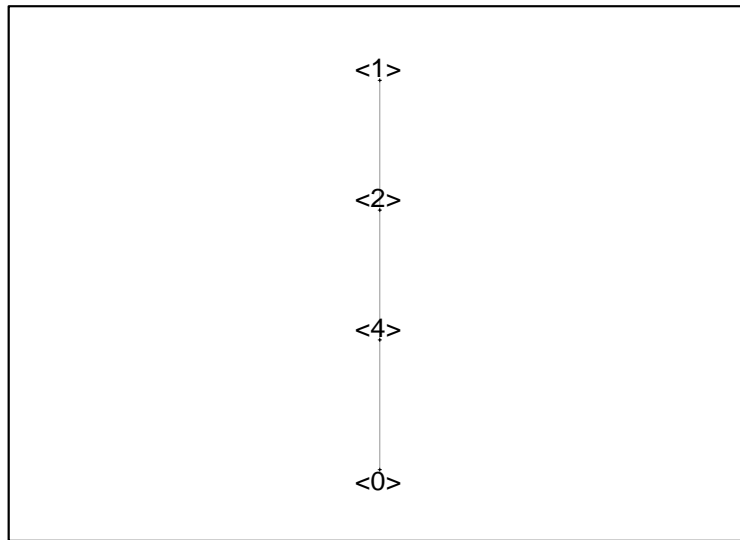
3. 7.

4. 12.

5. 57175109127133391183231447607307439381895.

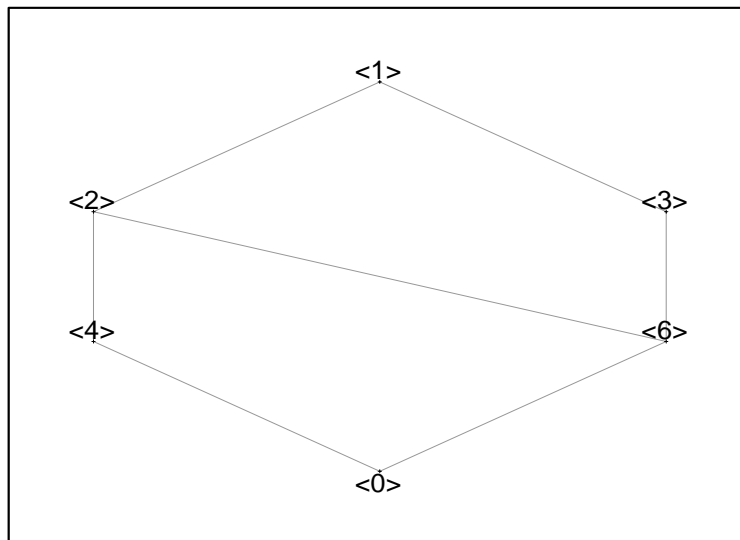
6.  $\left[ \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 2 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 1 & 0 \end{bmatrix} \right]$

7.  $Z_8$ :

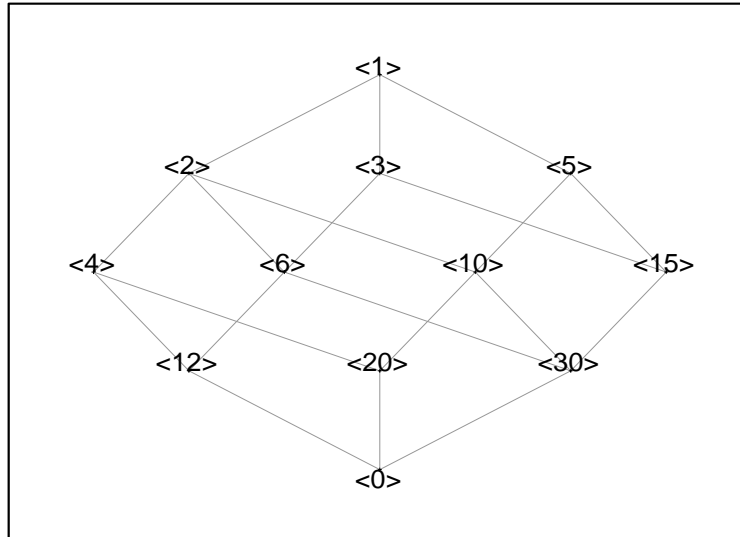


$Z_{12}$

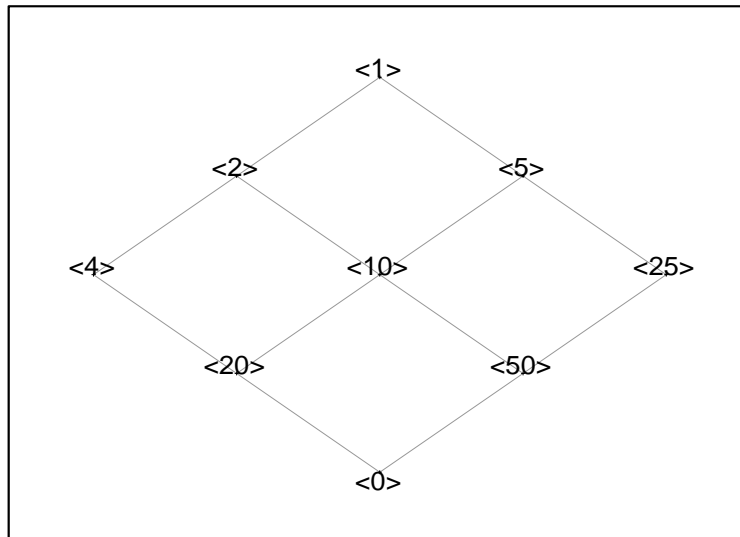
:



$Z_{60}$   
:



$Z_{100}$   
:



### 9.7 Supplement for Chapters 1 - 4

1.  $\{1, 5, 7, 11, 25, 29, 31, 35, 49, 53, 55, 59, 73, 77, 79, 83\}$ .

2.  $\left\{ \begin{bmatrix} 0 & 3 \\ 3 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 1 \\ 3 & 0 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}, \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 3 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \right\}$

