

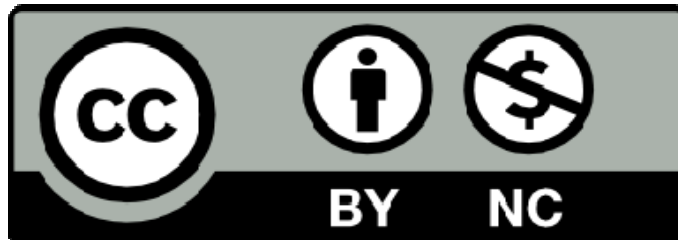
A CHILD'S GARDEN OF GROUPS

Frieze Groups, Wallpaper Groups, and Free
Range Symmetry

(Part 6)



by
Doc Benton



Creative Commons License

You are free to:

- Share this work
- Adapt this work
- Attribute all original materials to Doc Benton

You are not free to:

- Charge money for this work; Knowledge is free!

CONTENTS (PART 6)

| | |
|-----------------------------------|----|
| Introduction (Part 6) | 1 |
| Frieze Groups | 2 |
| Wallpaper Groups | 13 |
| Free Range Symmetry | 59 |
| How to Use GAP (Part 6) | 62 |
| Summary (Part 6) | 79 |
| Practice (Part 6) | 80 |
| Practice (Part 6) – Answers | 87 |

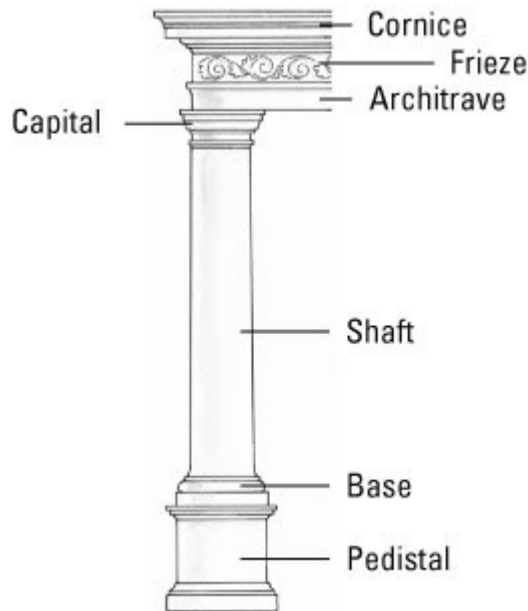
INTRODUCTION (PART 6)

In Part 6 we continue to explore some of the more visual aspects of *group theory*. In particular, some easy to use flowcharts are provided to help you identity the symmetries associated with friezes (the decorative patterns often found at the top of ancient Greek columns) and the symmetries associated with the various patterns found on wallpaper designs. Additionally, I introduce you to a less rigid way of identifying and analyzing symmetry that I like to call *free range symmetry*. No new commands for GAP will be introduced in Part 6, but as we've done previously, we'll include for reference all the GAP commands we've given up to this point.

FRIEZE GROUPS

Early on we mentioned that *groups* measure symmetry and that wherever we find symmetry present, we also find a corresponding *group*. We've also stated that symmetry is just the repetition of a pattern in some form or fashion and that the maneuvers that convert one instance of that pattern into another combine to form the elements of the associated *group*. We're now going to return to symmetry, and we begin with an introduction to *frieze groups*.

A frieze is the decorative strip that may adorn the edge of a column or building, and it is most often associated with Greek architecture.



Mathematicians have studied and classified the various types of symmetrical patterns that can occur in a frieze, and, as a result, seven different patterns have been identified, and they all involve a mixture of *translations*, *reflections*, and *rotations*. In this lesson we are going to explain and give examples of each of

the seven patterns. Also, the notation that was originally developed by mathematicians for these patterns was anything but intuitive. Thus, I am going to use a more modern and self-explanatory notation introduced by the British mathematician, John Conway.

F1-HOP: The *hop* is a frieze pattern that is generated solely by *translation* without any *reflections* or *rotations*. Below is a characterization of this pattern using a footstep along with a real world example.



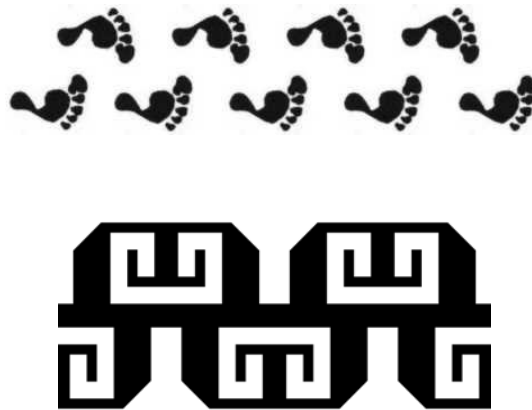
In classifying a frieze pattern, we need to be aware that they all contain symmetry or repetition based upon the horizontal movement or translation of a particular motif, and we also need to examine the image for reflections and rotational symmetries. Additionally, we want to identify a smallest possible piece of our frieze that could be used to generate the whole image. In the above example, it appears that the entire image can be translated by shifting the image in the cell below either left or right.




In this case, the corresponding *group* is going to be *isomorphic* to the *integers* under addition, $(\mathbb{Z}, +)$. In particular in this *group*, 0 corresponds to no movement at all,

1 corresponds to moving one unit to the right, and -1 represents moving the image one unit to the left.

F2-STEP: The *step* is also known as a *glide reflection*. This is a type of *reflection* that begins with a *translation* that is immediately followed by a *reflection*, usually across a horizontal line.



To generate the first image, we can think of taking a single foot print, , sliding it to the right, and then reflecting it to get our next foot print,



. The *group* created is also *isomorphic* to the *integers* under addition, $(\mathbb{Z}, +)$, where 1 means “move to the right and flip” while -1 means “move to the left and flip.” But in spite of getting back the same *group* associated with F1, the patterns, F1 and F2, are still distinct since F1 involves only a *translation* while F2 involves both a *translation* and a *flip*.

F3-SIDLE: The word *sidle* can mean to walk sideways, and the image below looks like what would happen if you stood with your feet turned out and then tried to walk sideways.



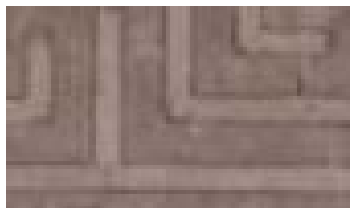
The *sidle* contains both a *translation* and a *reflection* about what is normally a vertical axis. If you look at the first image above, you can see how to create F3, a *sidle*. You can start with the imprint of a single right foot, and then reflect it about a vertical axis. We usually think of using the same, fixed vertical axis for all reflections. Next, just translate both foot prints left or right to complete the frieze. The presence of both *translations* and *vertical reflections* in the second image is an immediate giveaway that the symmetry pattern is F3, the *sidle*.

F4-SPINNING HOP: The *spinning hop* combines a *translation* with a *rotation* through 180° . Notice that you can also accomplish this half-turn by doing a *reflection* about a horizontal axis that is immediately followed by a *reflection* across a vertical axis. In the image below, you can start with a single footprint with the toes on the left, reflect it about a horizontal and then a vertical axis to get a second footprint, and then translate both feet right or left to complete things.

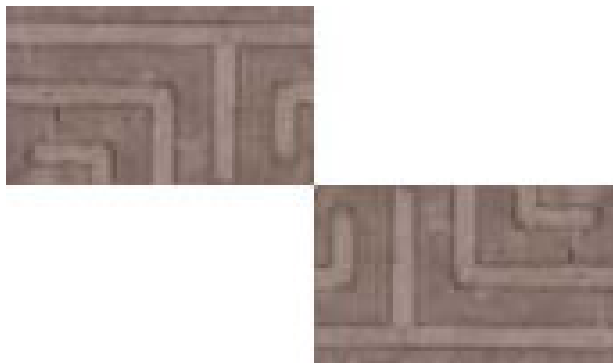




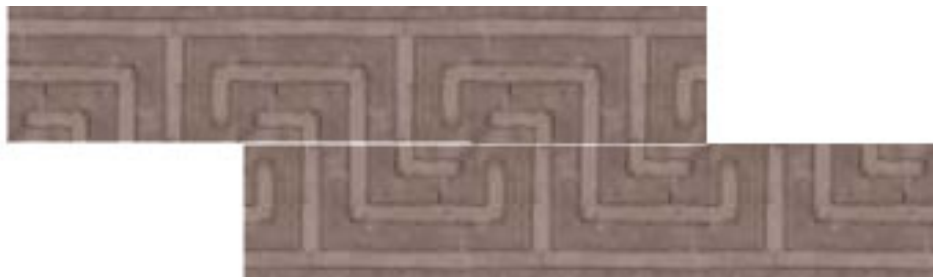
To generate this image, I can start with a rectangle that extends from one center of rotation to the next.



Next, we rotate this piece 180° about the upper left corner.



And now, we can begin to complete the image by doing translations right and left.



F5-SPINNING SIDLE: This particular frieze pattern, the *spinning sidle*, exhibits *translations*, *glide reflections*, *reflections* about a vertical axis, and *rotations* through 180° .



Even though this frieze pattern exhibits multiple symmetries, the whole frieze can be generated by taking an image representing the fundamental pattern and then applying *reflections* about a vertical axis and *glide reflections* as well. For example, we can start with the following image.



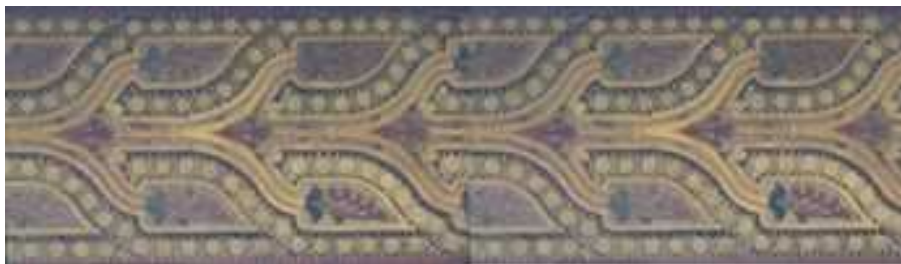
Now *reflect* this image about a vertical axis.



And lastly, do *glide reflections*.



F6-JUMP: This pattern, the *jump*, is exactly what you would get if you stood on two feet and hopped in a consistent direction. The image consists of a translation coupled with a horizontal reflection.



We can generate this type of frieze by starting first with a fundamental region.



Next, we *reflect* this piece about a horizontal axis.



And to complete the image, we do a *translation* of this image right and left.



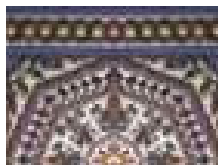
F7-SPINNING JUMP: The *spinning jump* combines all possible maneuvers. In other words, it contains *translations*, *glide reflections*, *vertical reflections*, *rotations*, and *horizontal reflections*. However, the image itself can be generated by applying just *translations*, *vertical reflections*, and *horizontal reflections* to a fundamental region.



So, let's start with a fundamental image that can be used to generate the rest of the pattern.



Do a *flip* about a vertical axis.



Follow this with a *flip* about a horizontal axis.



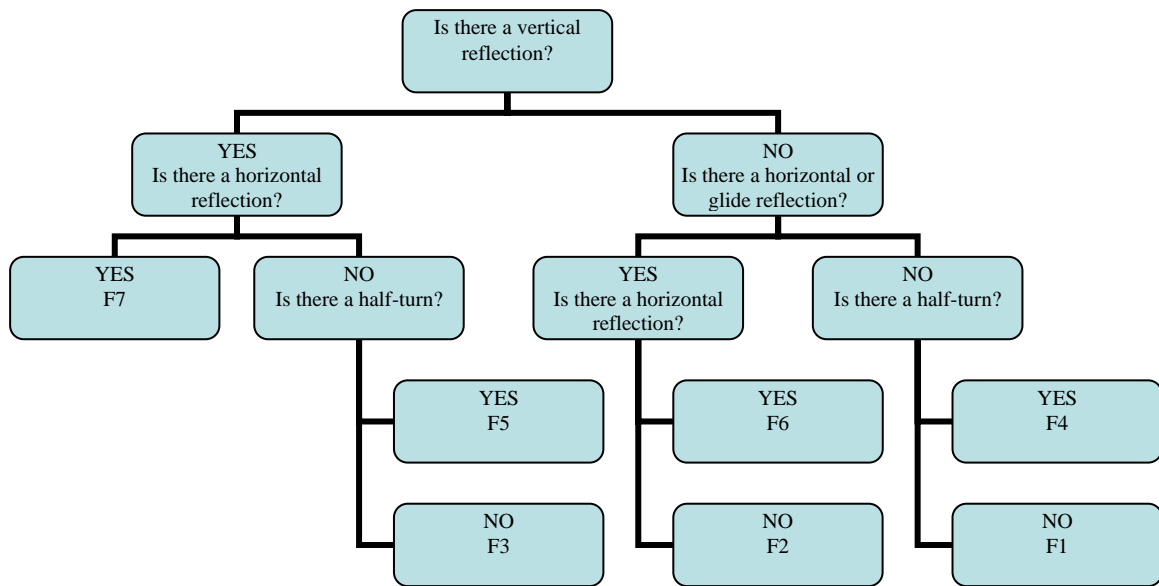
And finally, translate this image left and right to finish generating the frieze.



By the way, in the image below, you might have also noticed the presence of *rotational symmetry* through angles of 45° , and you can rightly wonder why this symmetry isn't part of our analysis. The answer is because in the symmetries that we associate with frieze groups, we restrict ourselves to rigid motions of the plane. For example, if we *rotated* the entire image clockwise through angles that are *integer* multiples of 45° , then we could get something like the following which deviates considerably from our frieze pattern.

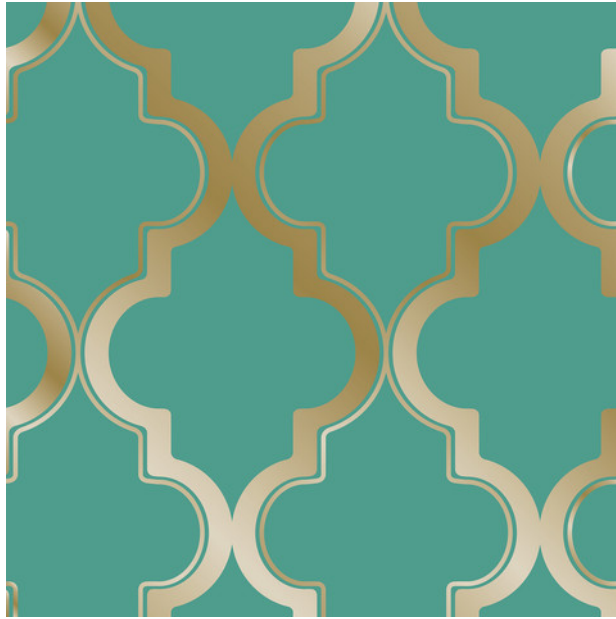


And finally, below is a nice flowchart that can help you identify which of the seven types of frieze groups you are observing. Enjoy!



WALLPAPER GROUPS

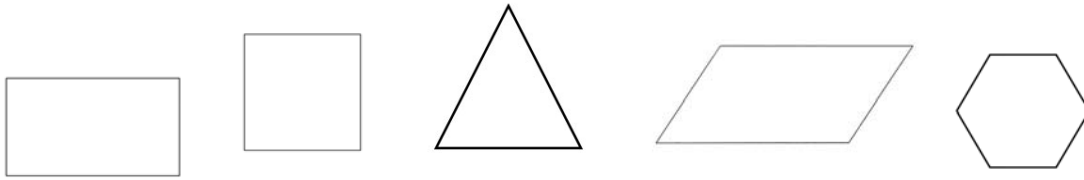
Related to *frieze groups* are *wallpaper groups*, and these are the *groups* that are defined by the symmetries found in many examples of wallpaper or floor tiles.



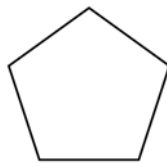
These symmetries involve *translations*, *reflections*, *glide reflections*, and *rotations* through various angles. Also, whereas there are only 7 distinct *frieze groups*, there are 17 distinct *wallpaper groups*, and if we extend ourselves into 3-dimensional space, there are 230 different *space groups*! We'll stop our explorations, however, with *wallpaper groups*.

As you might imagine, there are some restrictions regarding the angles that give rise to *rotational symmetry* in a wallpaper pattern. This is because while you can tile a plane with shapes such as a rectangle, square, equilateral triangle, parallelogram, or regular hexagon, you can't tile a plane with a regular pentagon. Hence, you won't find rotations through angles of $360^\circ / 5 = 72^\circ$ in a wallpaper or floor tile pattern. Instead, the permissible rotations are 0° (none), 60° , 90° , 120° , or 180° .

These shapes can be used as floor tiles



This shape can't be used as a floor tile



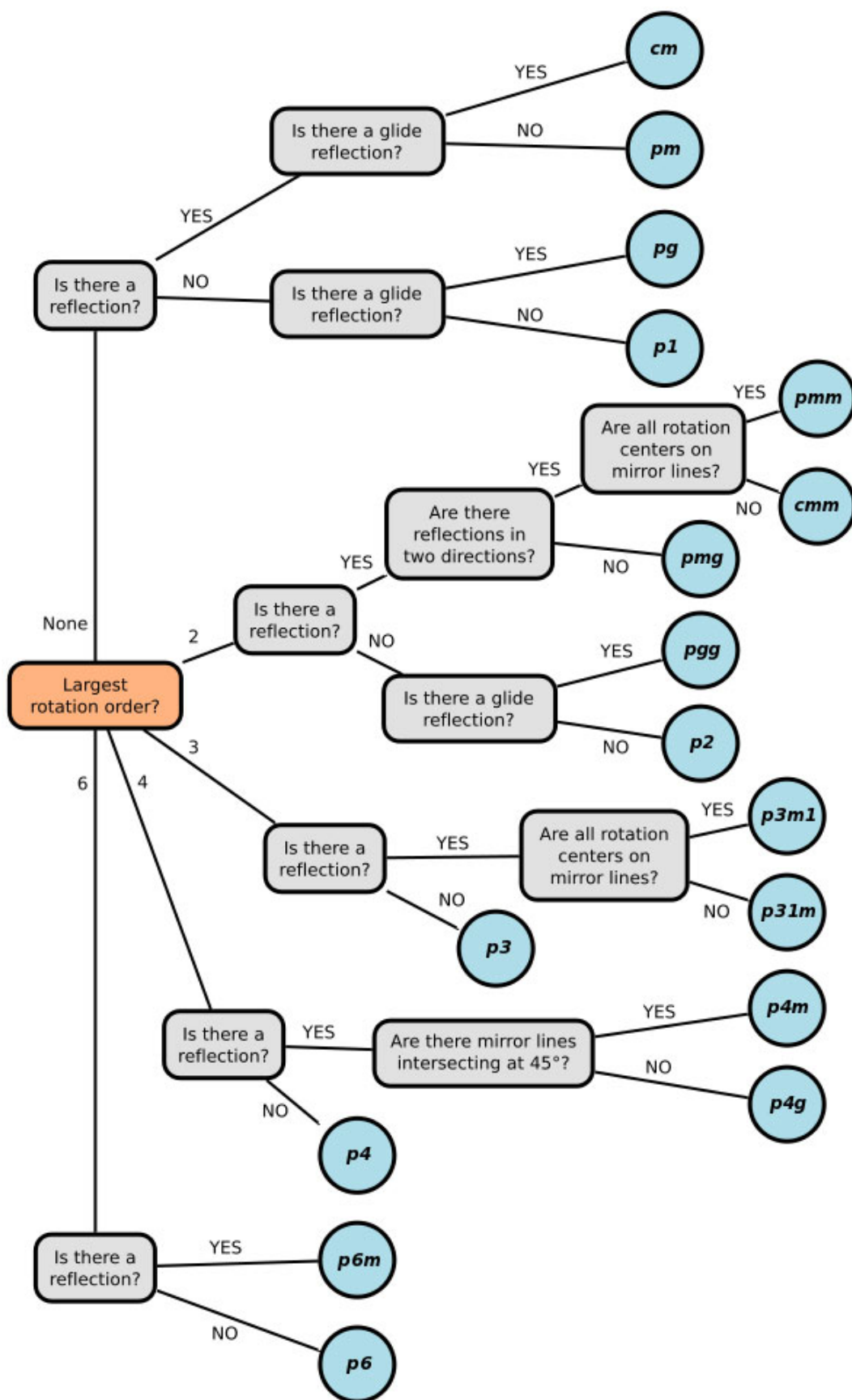
Below is a flowchart for helping you classify a *wallpaper group*. You begin by deciding what kind of *rotational symmetry* is present, and then you study the image for *reflections*. If, for example, you see rotations through angles of 180° , then that means the order of the rotation is 2 since $\frac{360^\circ}{180^\circ} = 2$. The notation for the different wallpaper patterns is not that intuitively obvious, but letters like *m* stand for *mirror lines*, *g* stands for *glide reflection*, and a number like 2 stands for *2-fold rotation*. For a fuller explanation of the notation for wallpaper groups, check out the article titled “Wallpaper Group” at the Wikipedia. Also, many of these wallpaper patterns can be found within the intriguing art of M.C. Escher, and so we include pictures from ten of his prints for you to study in order to gain some facility in using the flowchart, and additional information on Escher and wallpaper patterns may be found at

http://euler.slu.edu/escher/index.php/Wallpaper_Patterns. Enjoy!

| Symmetry Group - IUC | Lattice | Rotation | Reflection |
|----------------------|---------------|----------|------------|
| p1 | Parallelogram | none | none |
| p2 | Parallelogram | 2 | none |
| pm | Rectangle | none | parallel |
| pg | Rectangle | none | none |
| cm | Rhombus | none | parallel |
| pmm | Rectangle | 2 | 90 degrees |
| pmg | Rectangle | 2 | parallel |
| pgg | Rectangle | 2 | none |
| cmm | Rhombus | 2 | 90 degrees |
| p4 | Square | 4 | none |
| p4m | Square | 4 + | none |
| p4g | Square | 4 * | 45 degrees |
| p3 | Hexagon | 3 | none |
| p31m | Hexagon | 3 * | 60 degrees |
| p3m1 | Hexagon | 3 + | 30 degrees |
| p6 | Hexagon | 6 | none |
| p6m | Hexagon | 6 | 30 degrees |

The Seventeen Wallpaper Patterns

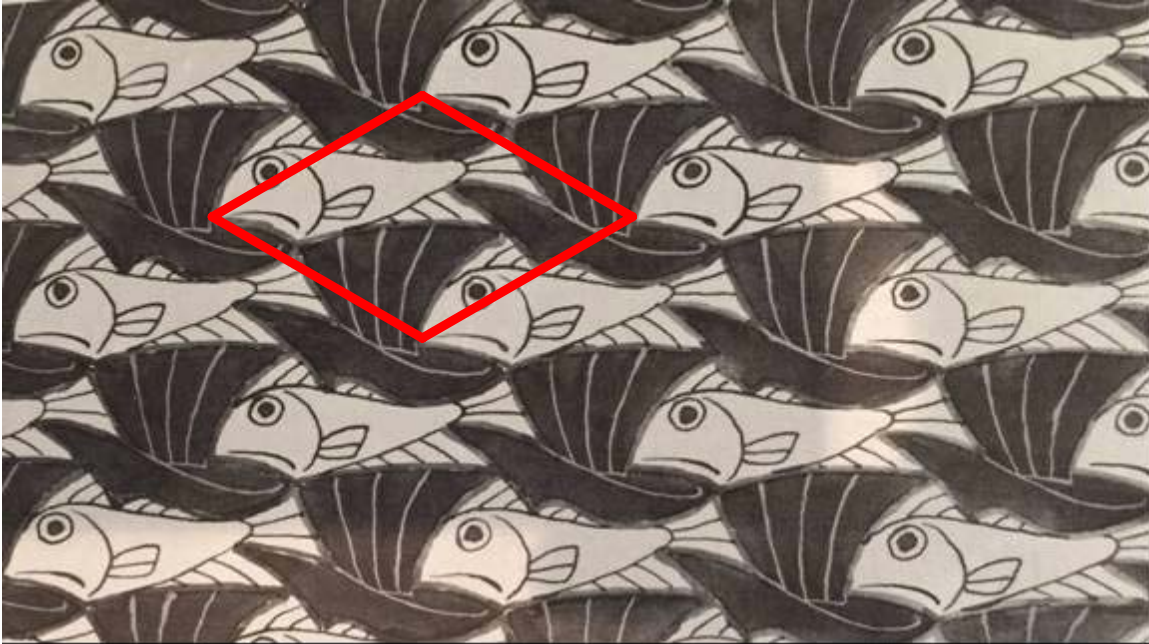
| Parallelogram | Rectangular | Rhombic | Square | Hexagonal |
|--------------------|---------------------|---------------------|---------------------|----------------------|
| p1 | pm | cm | p4 | p3 |
| p2 | pg | cmm | p4m | p3m1 |
| | pmm | | p4g | p31m |
| | pmg | | | p6 |
| | pgg | | | p6m |



p1

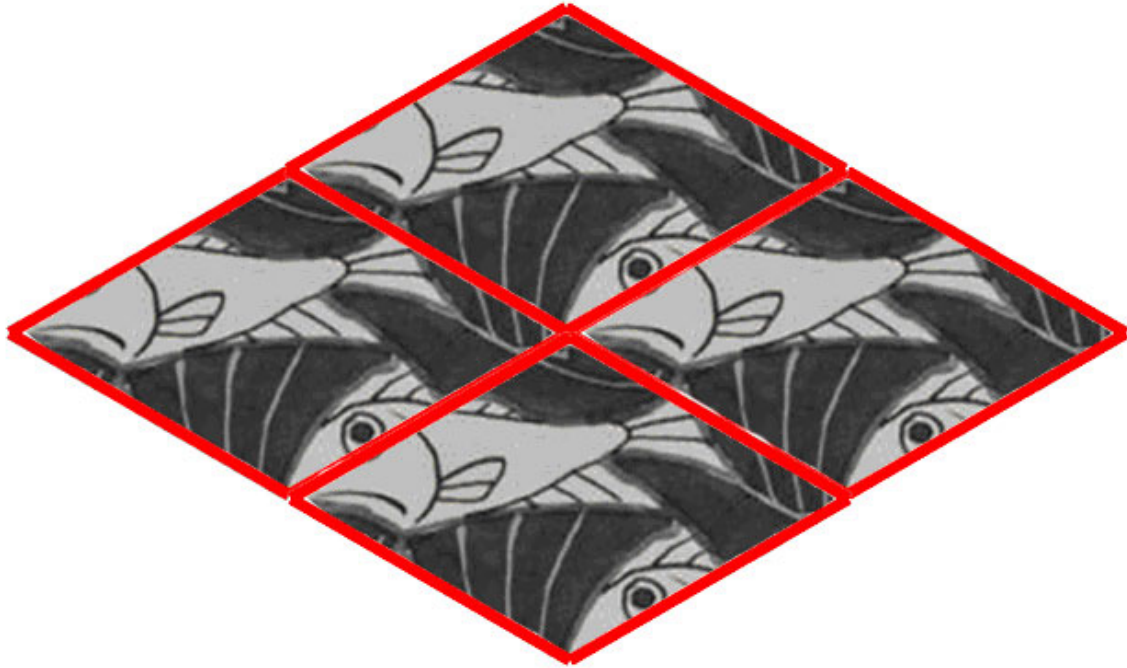


The simplest type of wallpaper pattern we can have is one in which there are no rotations, no reflections, and no glide reflections, and this is the $p1$ group that consists only of translations or repetitions of a basic pattern in two different directions. That is the type of wallpaper group illustrated by the image above.



Since there are no rotations or (glide) reflections in the above image, we can define a fundamental region whose translations generate the design by doing what I've done above. In other words, take a particular point such as the tip of the fishes snout, and connect that with other occurrences of that point in order to define your fundamental region that can generate the entire image through translations. This gives us the following parallelogram.

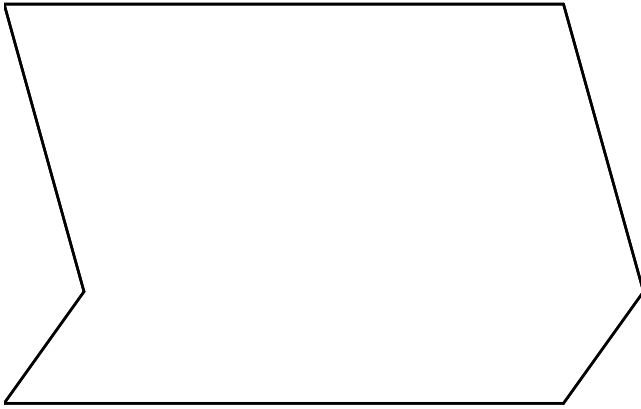




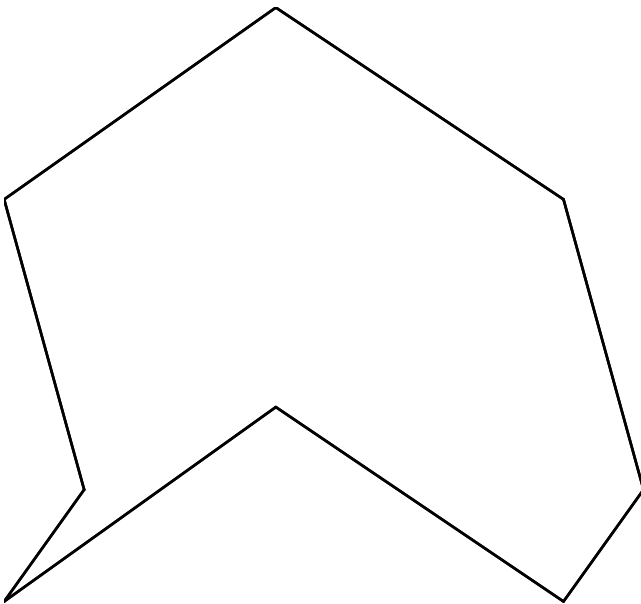
Now observe how we can start with an image such as a rectangle and change it into a non-rectangular fundamental region by simply altering the left and right side or the top and bottom sides in similar ways.



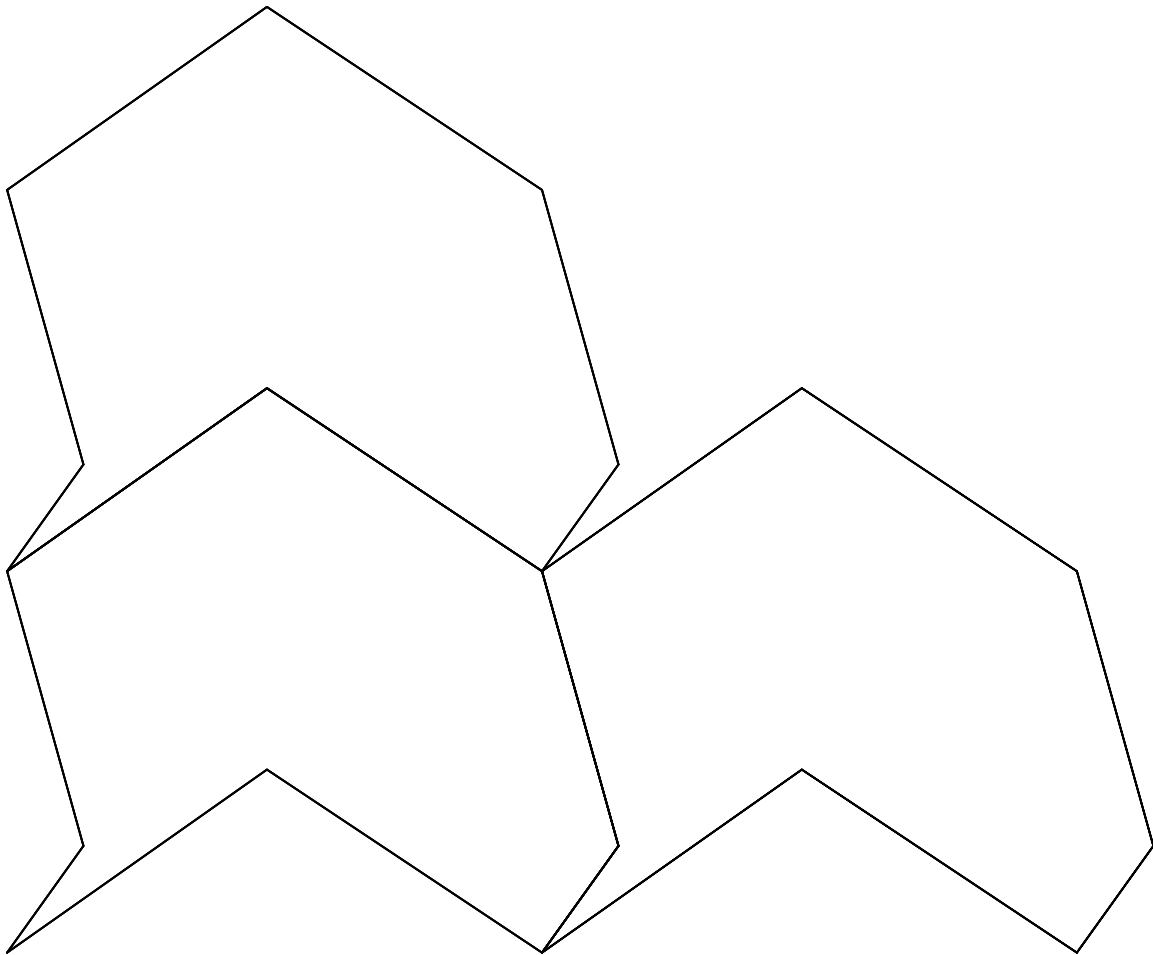
For example, here I have altered just the left and right sides of the rectangle.



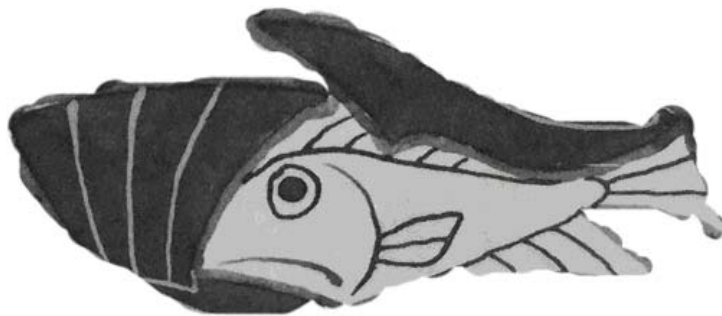
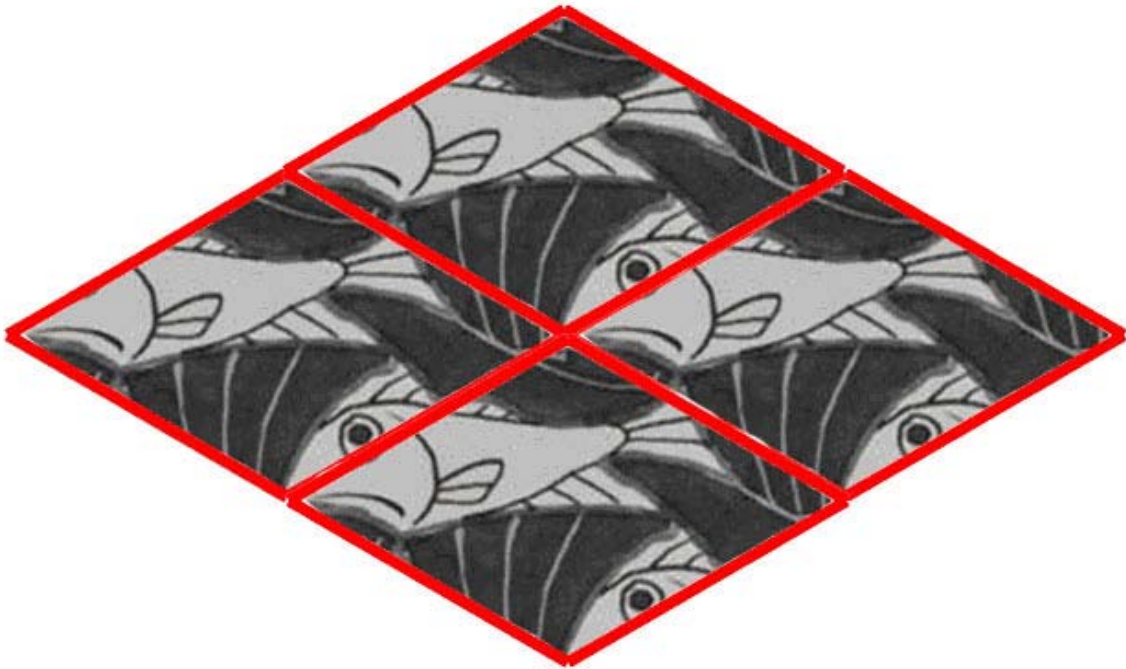
And now I've altered both the left and the right sides and the top and the bottom sides in corresponding ways.

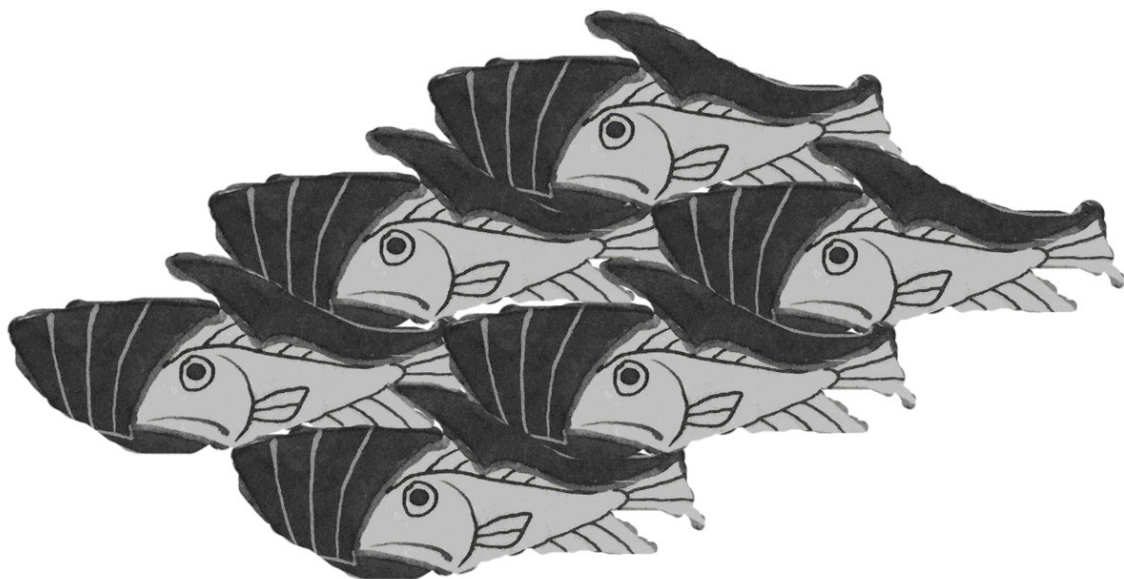


And now, copies of my last altered rectangle can be pieced together in order to generate the entire wallpaper pattern.



For example, if we apply this technique to one of the parallelograms below, then we can obtain a more recognizable “fish” motif that can be used to generate the entire wallpaper.





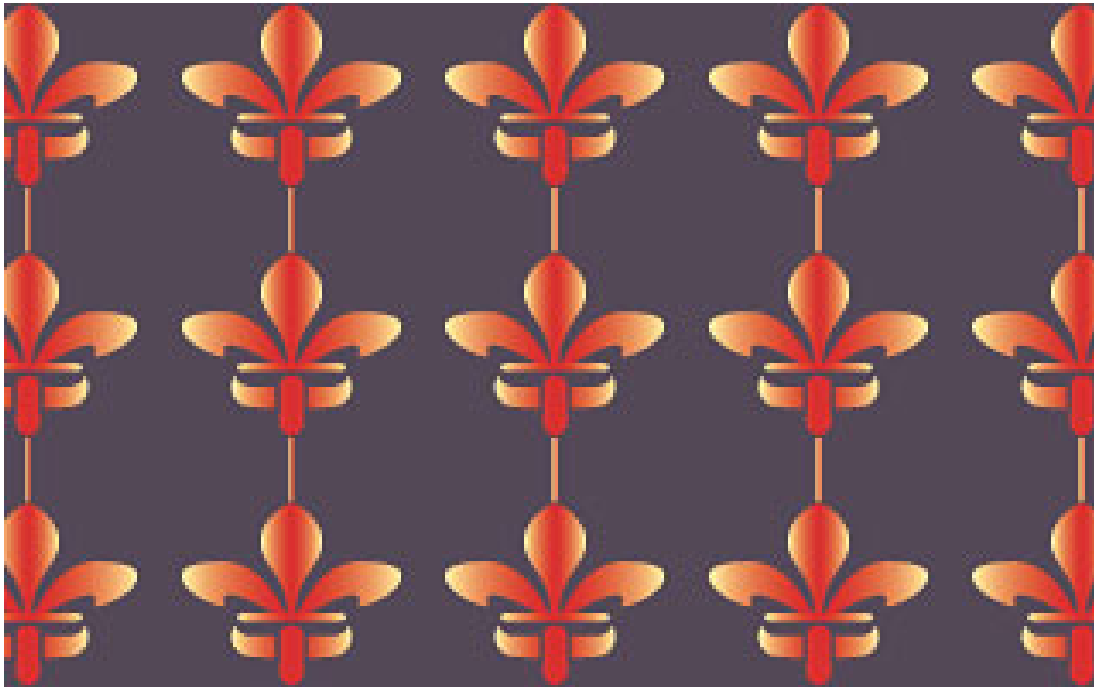
pg



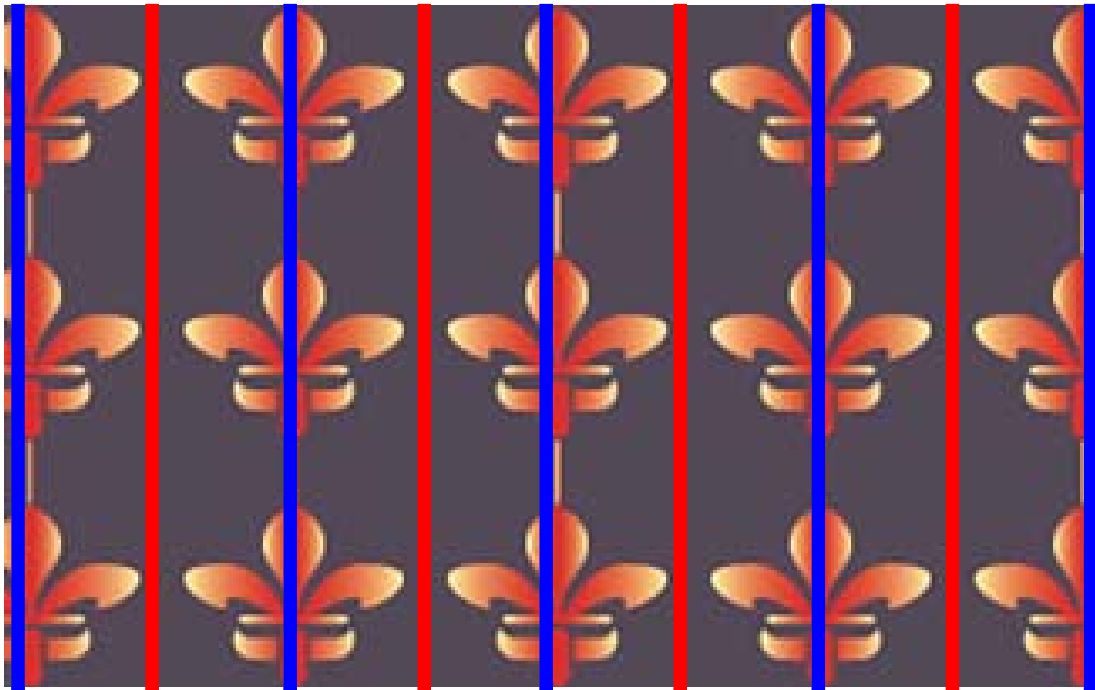
If there are no rotations and no reflections, but there is a glide reflection, then the corresponding wallpaper group is designated as pg and it is the glide reflection that distinguishes this wallpaper group from $p1$. Below I've marked two sets of axes for glide reflections. One set is marked in red and the other in a shade of green, and I'll always denote glide reflections by dashed rather than solid lines. There may be more glide reflections, but that's all I've found!



pm



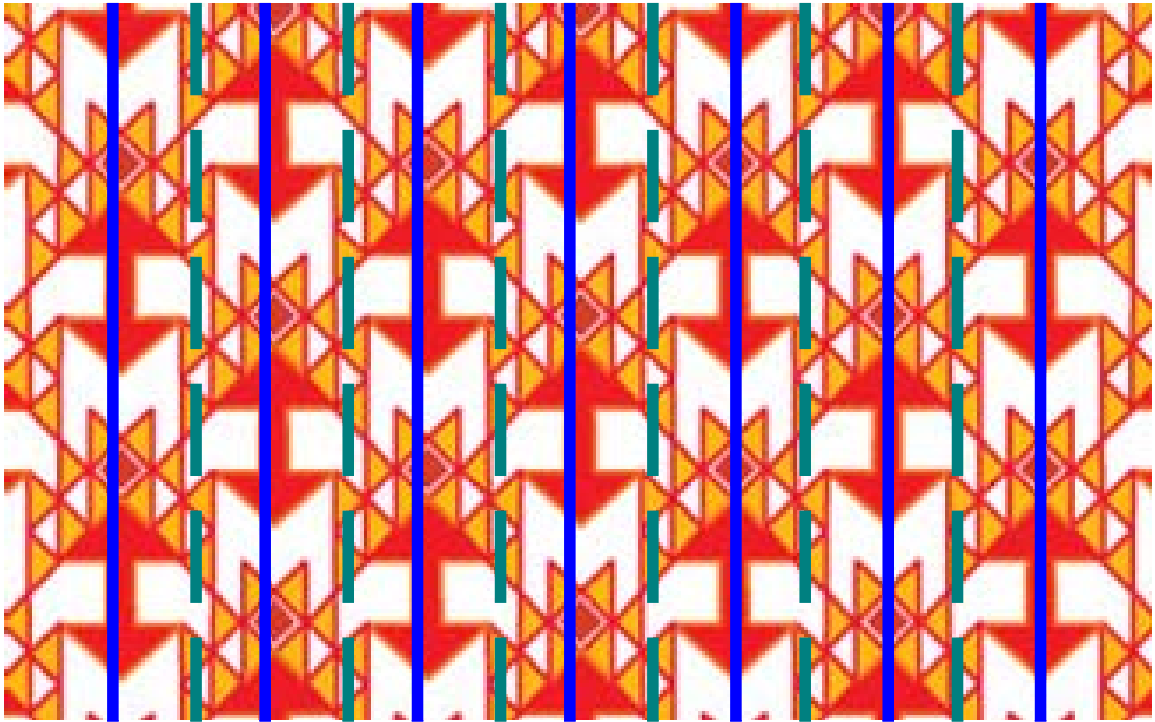
The next simplest wallpaper group is designated by pm . This pattern contains no rotations nor glide reflections, but it does contain at least one reflection. Below I've used colored solid lines, red and blue, to denote the two reflections axes that I see. Also, note that some of my reflection lines may be drawn slightly off target, and that is due to the limitations of the software program that I am using. Deal with it!

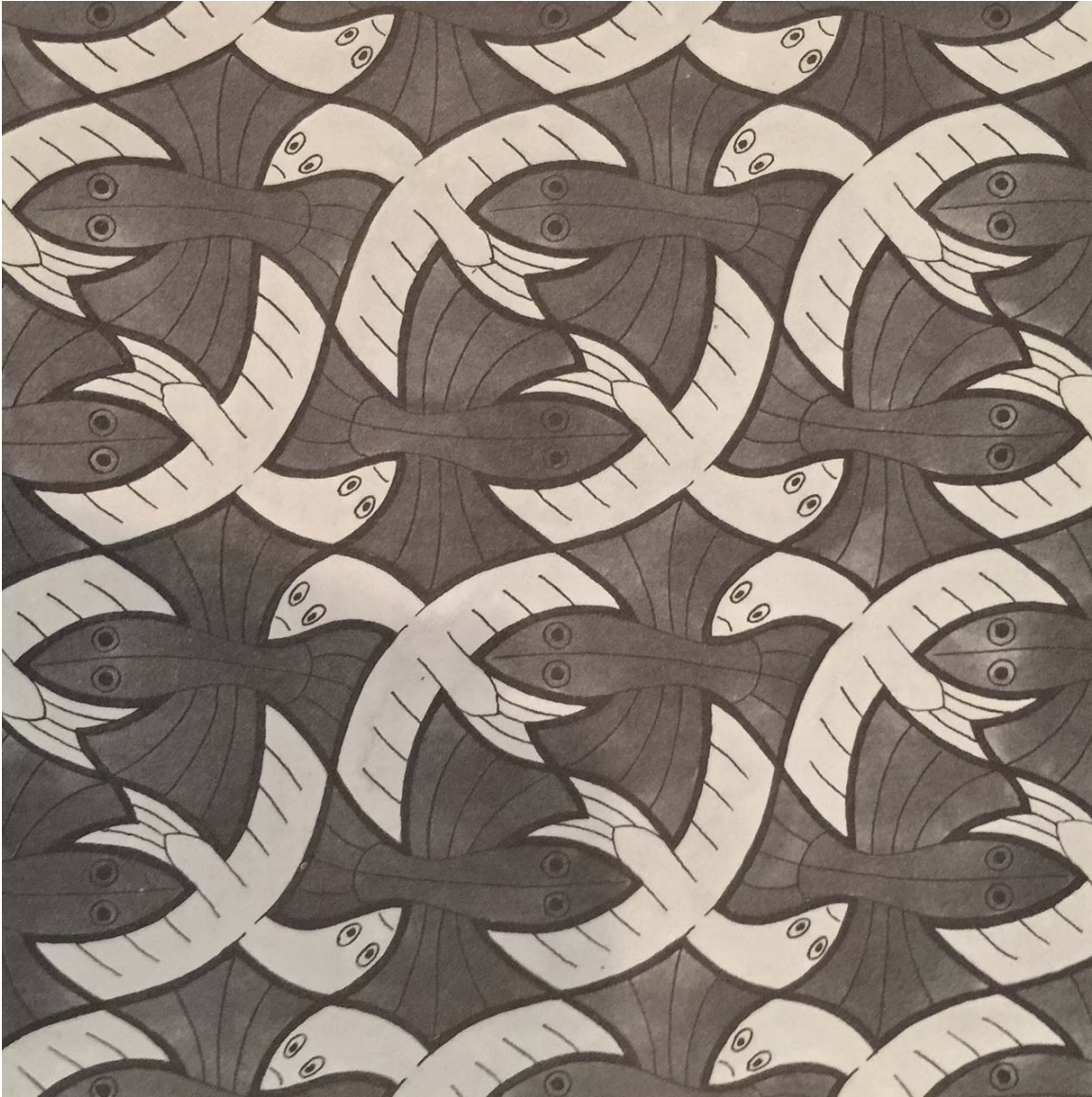


Cm

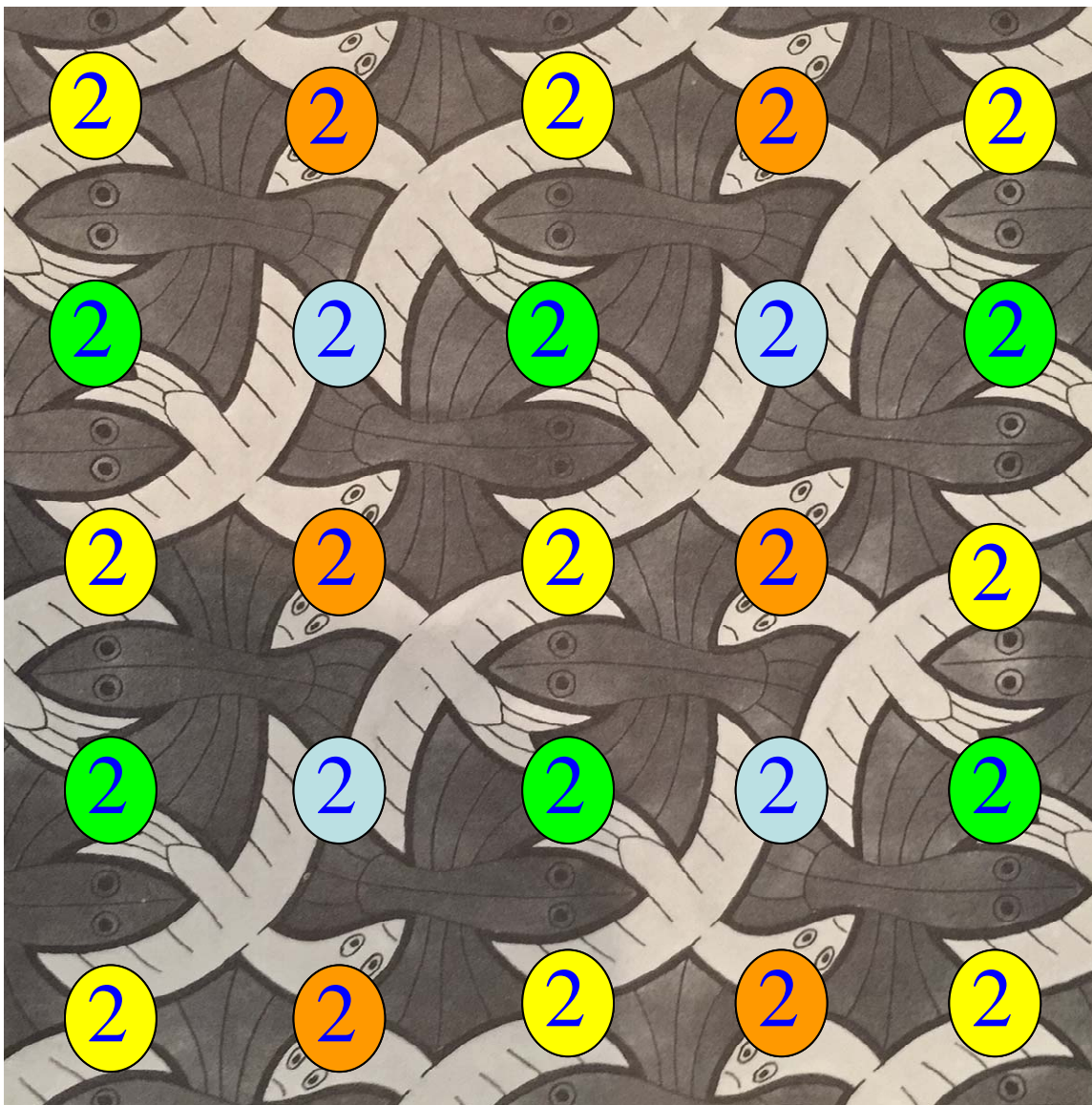


Next in complexity is a wallpaper design with no rotations, but both reflections and glide reflections. This wallpaper group is designated by *cm*. In the picture below I've marked the reflections I've found as solid lines and the glide reflections as dashed lines. Notice, too, that axes for glide reflections tend to be found halfway between axes for reflections.



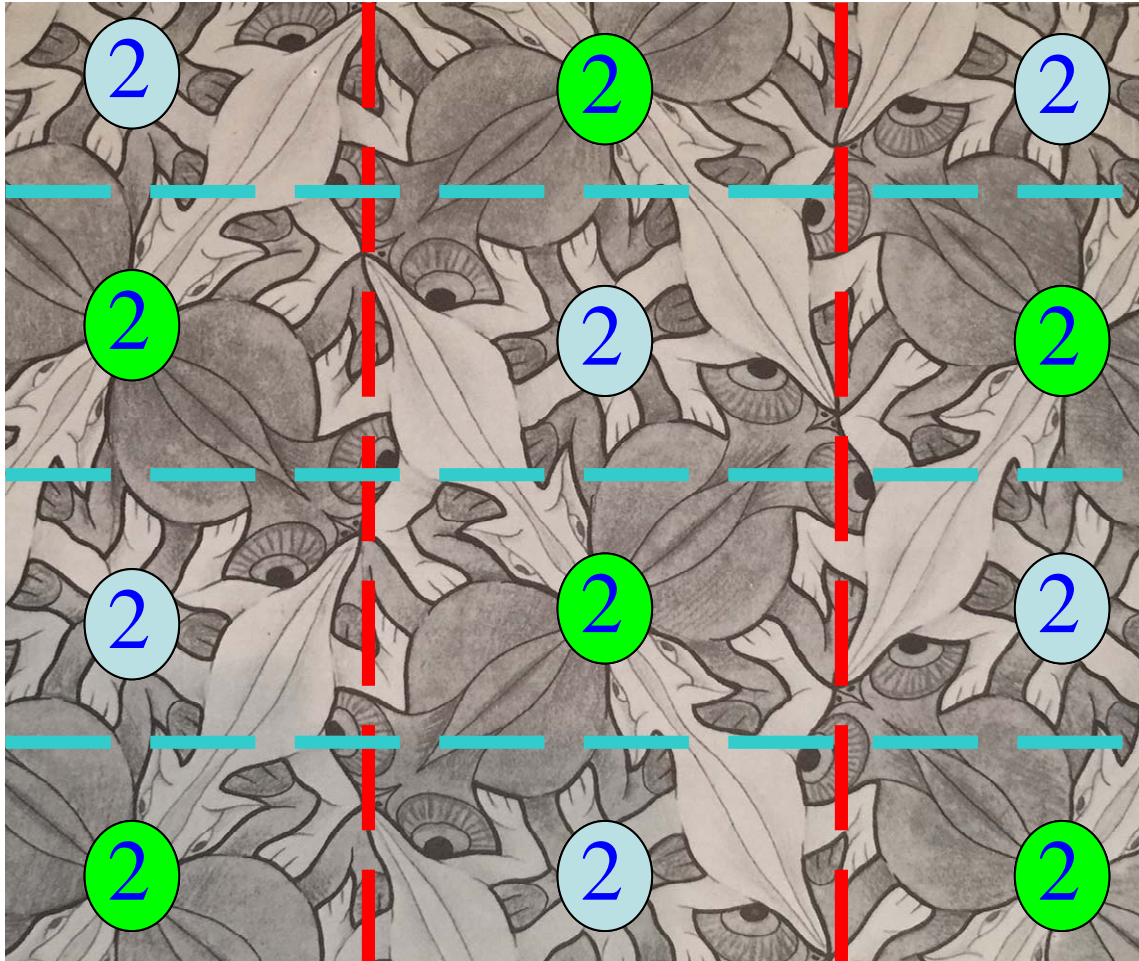


This wallpaper pattern has no reflections and no glide reflections, but it does contain rotations of order 2. That is, rotations through angles of 180 degrees. In the image below I've marked all the centers of rotation of order 2 that I've found, and I've color coded them. Basically, I found centers of rotation where the heads touch, where the tail feathers touch, where the left wings touch, and where the right wings touch.





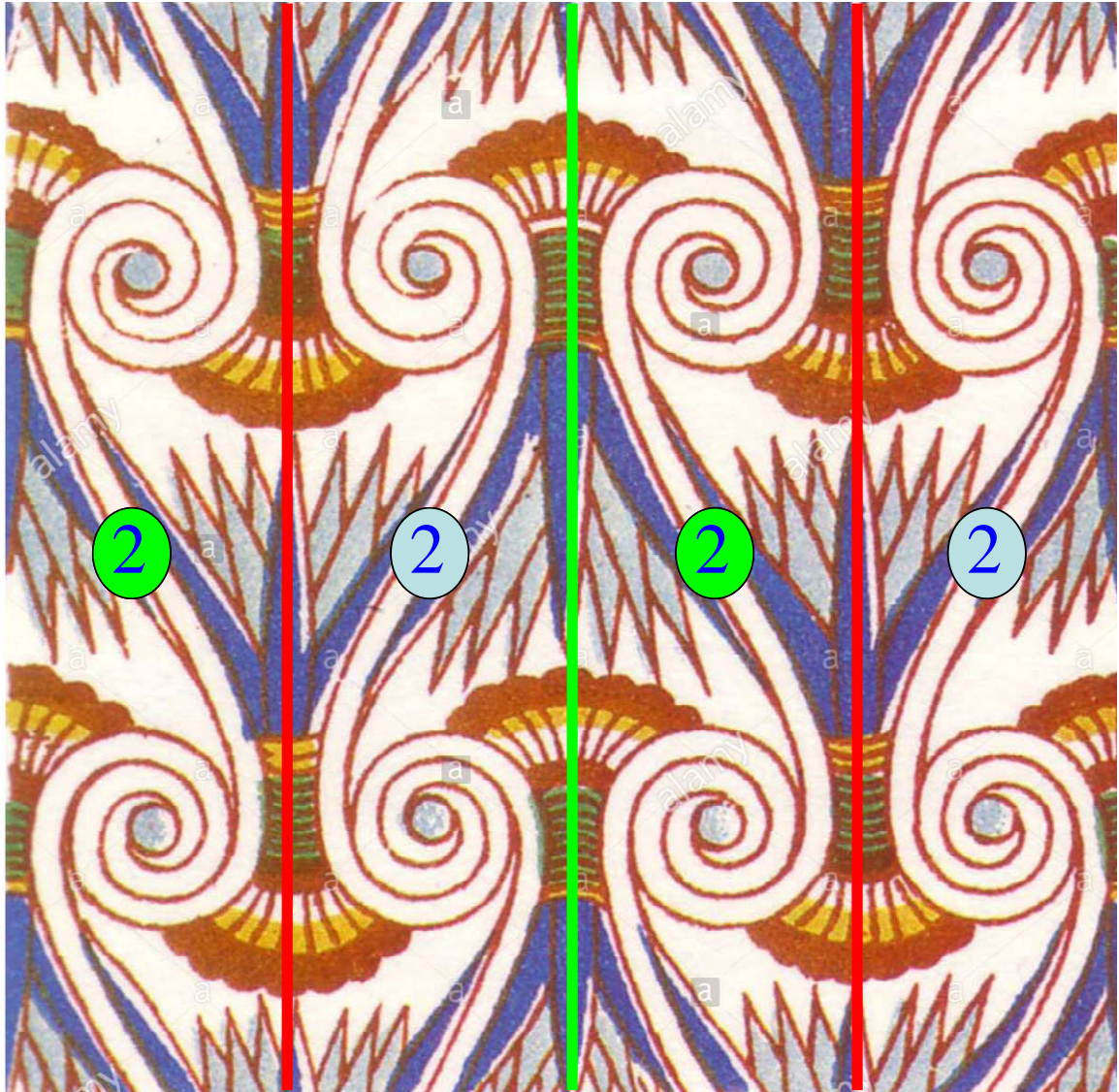
This wallpaper pattern contains rotations are order 2, no mirror lines, but some glide reflections. Below is the image marked with the centers of rotation and glide reflections that I found.



pmg



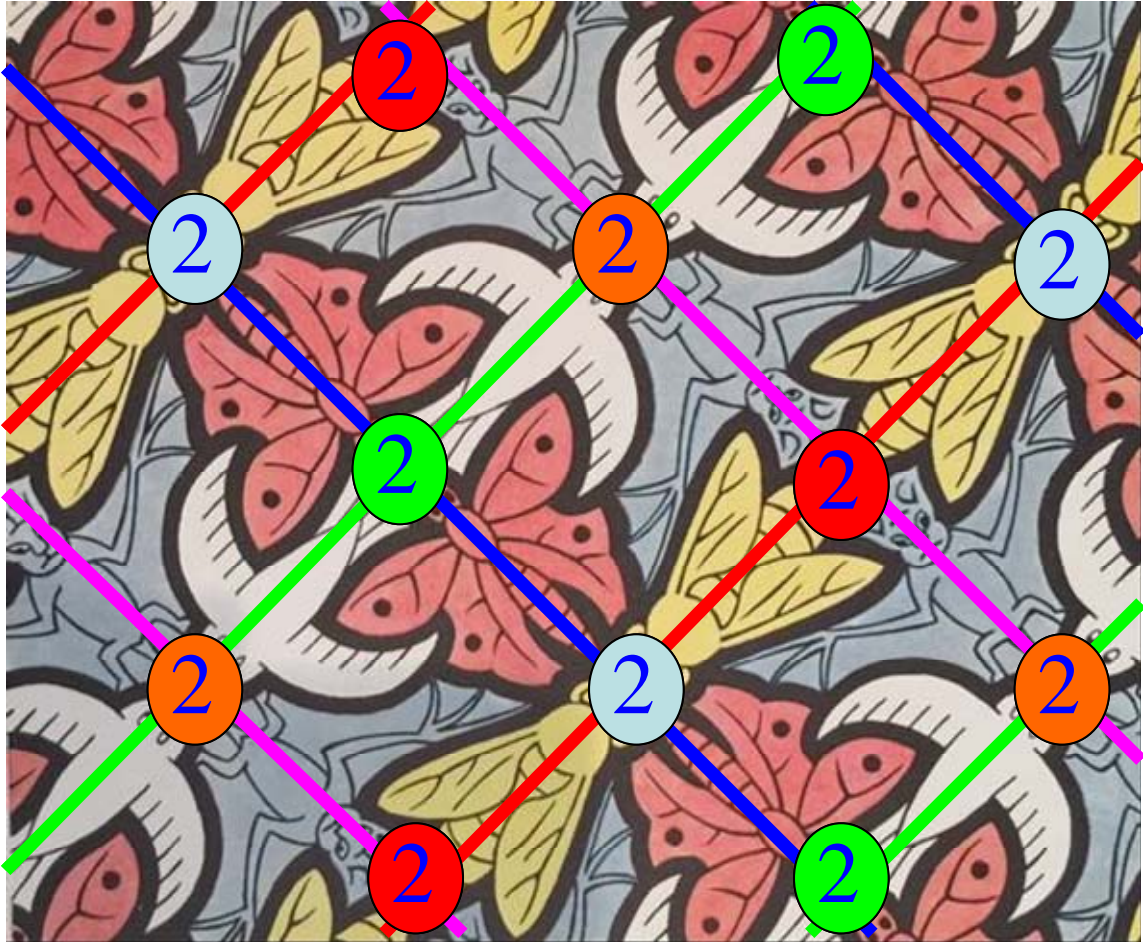
This wallpaper pattern contains both mirror lines and rotations through 180 degrees, i.e. rotations of order 2. Also, the mirror lines are all parallel to one another. Below I've marked the features I found in this design.



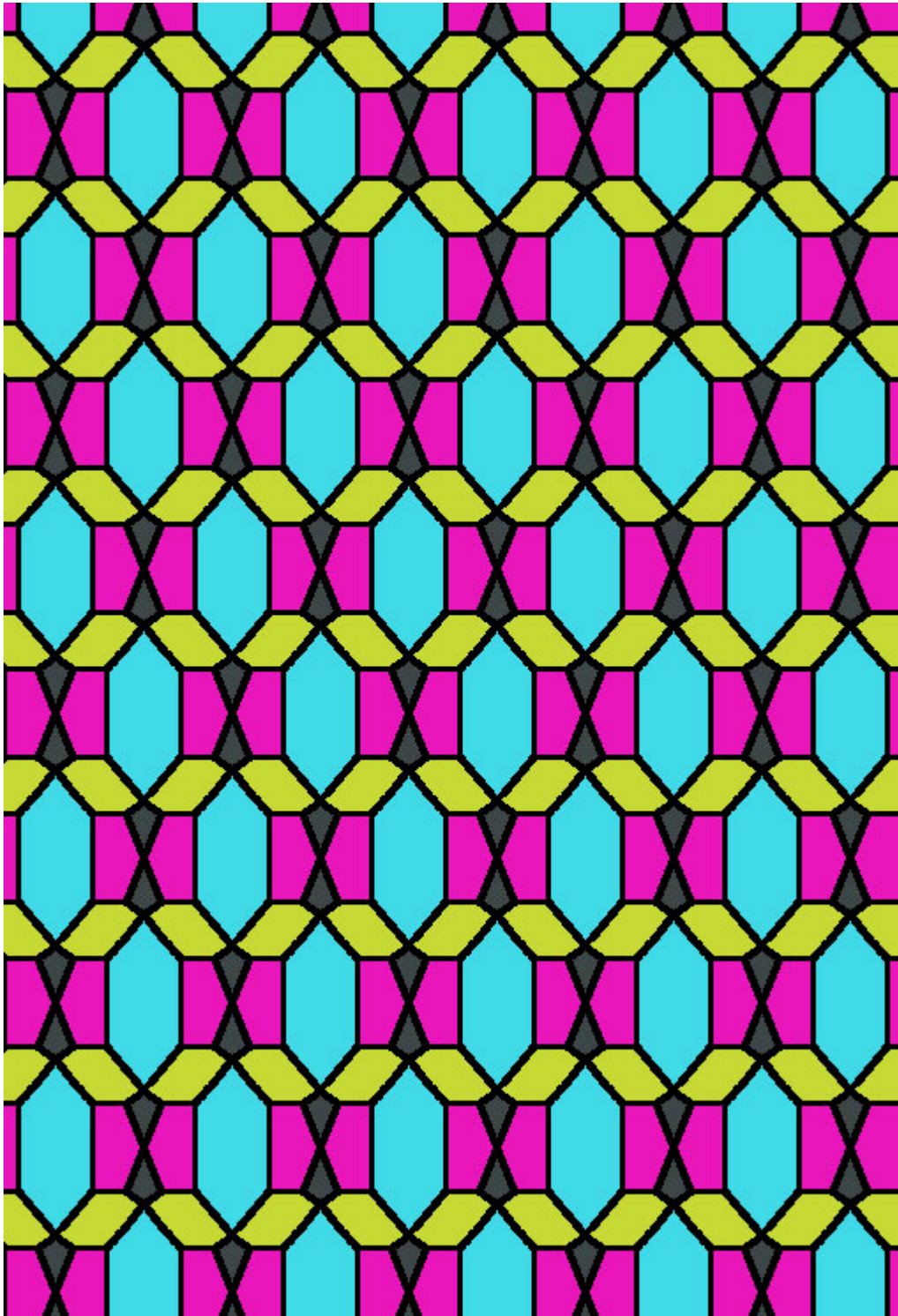
pmm



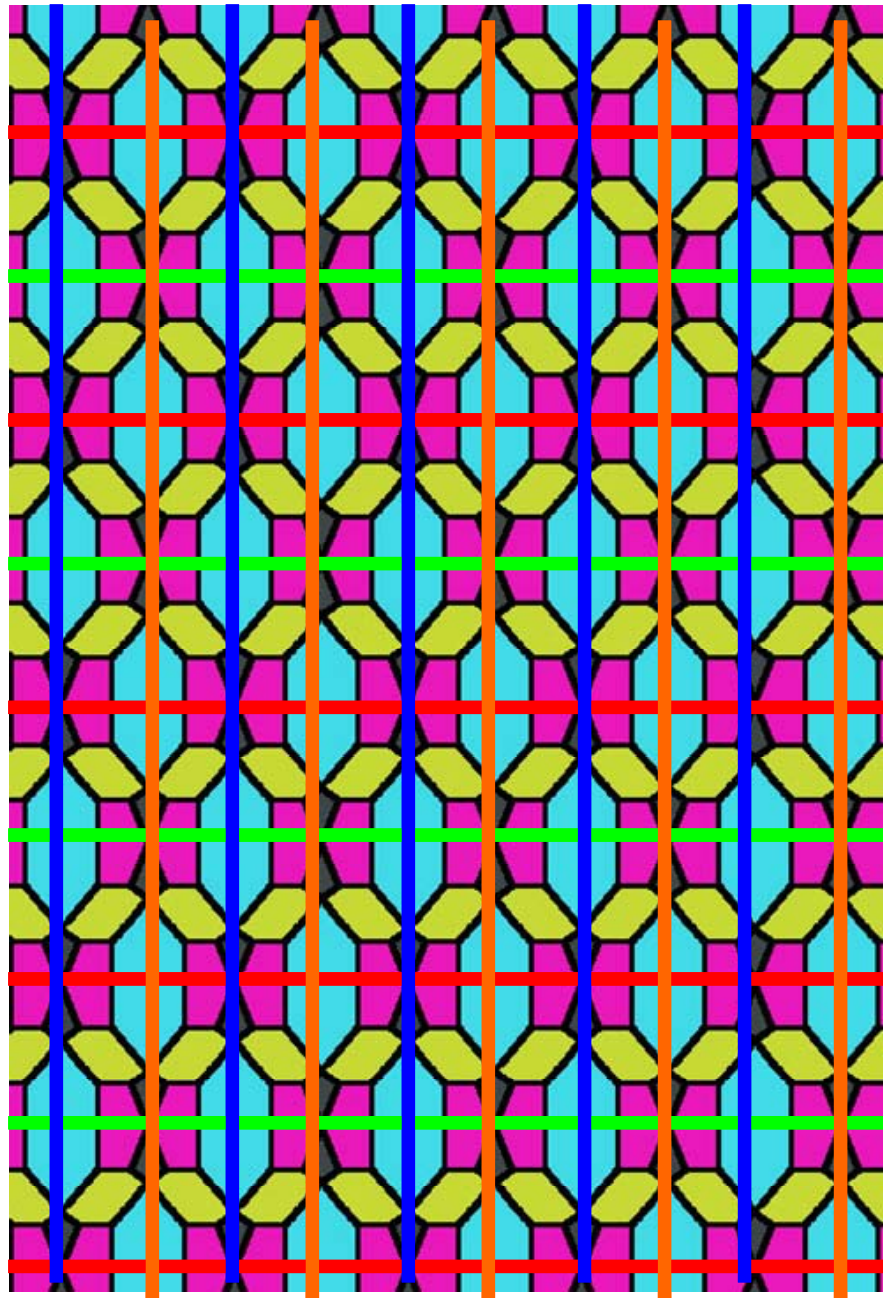
This is the first of two remaining patterns with rotations of order 2. This pattern also has mirror lines, but this time they do not all point in the same direction. Additionally, all centers of rotation will be found on the mirror lines.



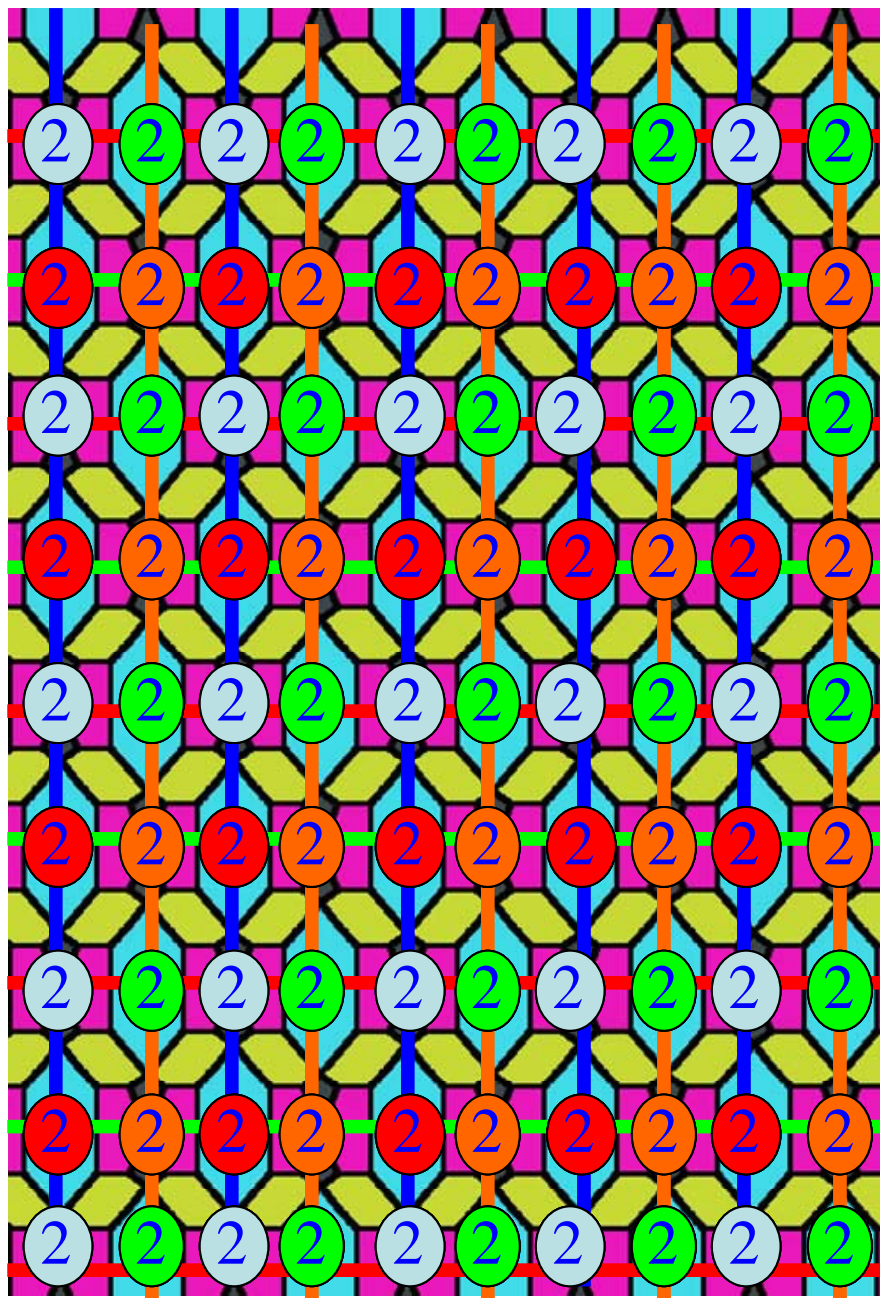
cmm



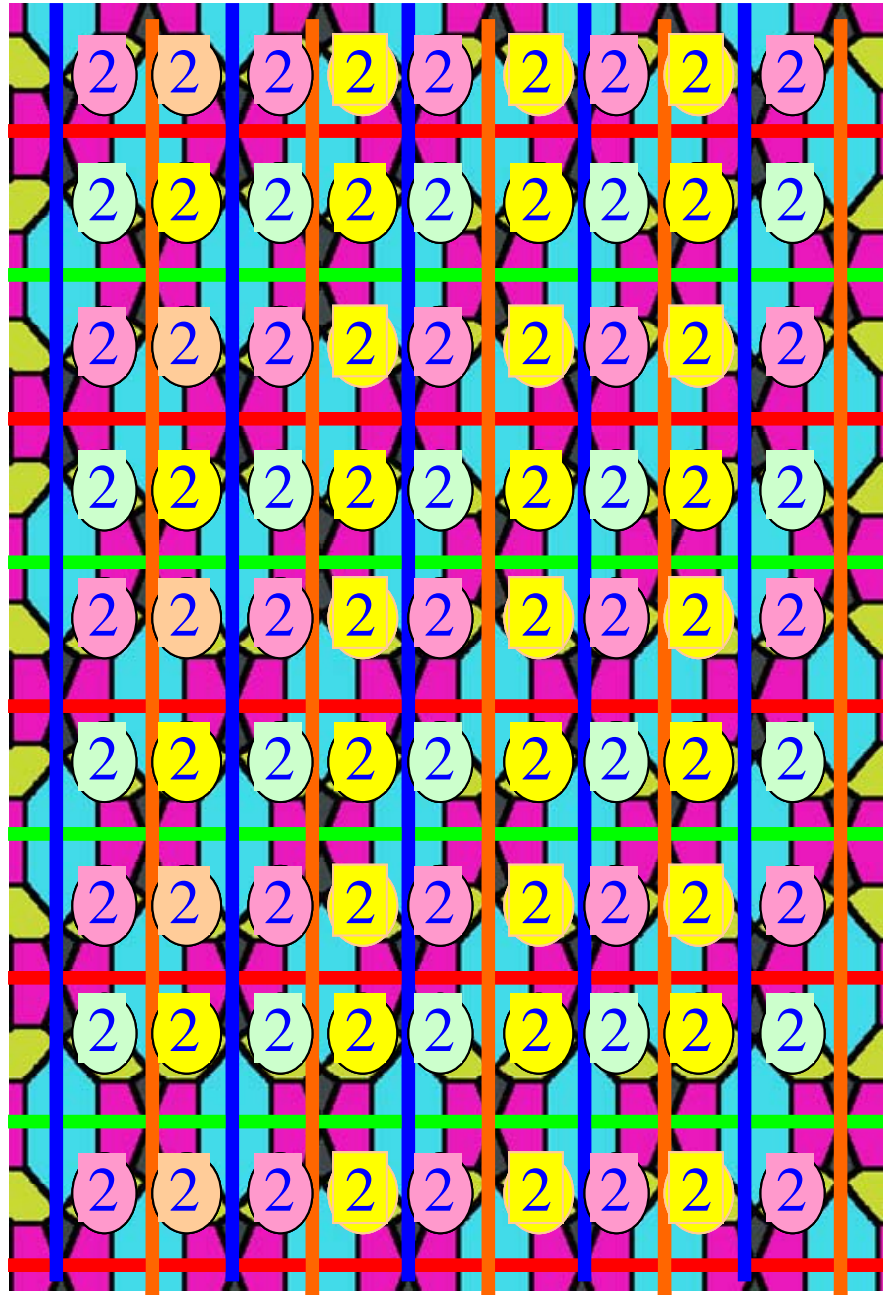
This wallpaper design contains rotations of order 2, reflection or mirror lines that are not parallel, and centers of rotation that are not all located on mirror lines. Below I've first marked the locations of all mirror lines that I found.



Next, I've added all the centers of rotations of order 2 that I found on mirror or reflection lines.



And lastly, I've marked all the centers of rotation I've found that are not on mirror lines.



p3



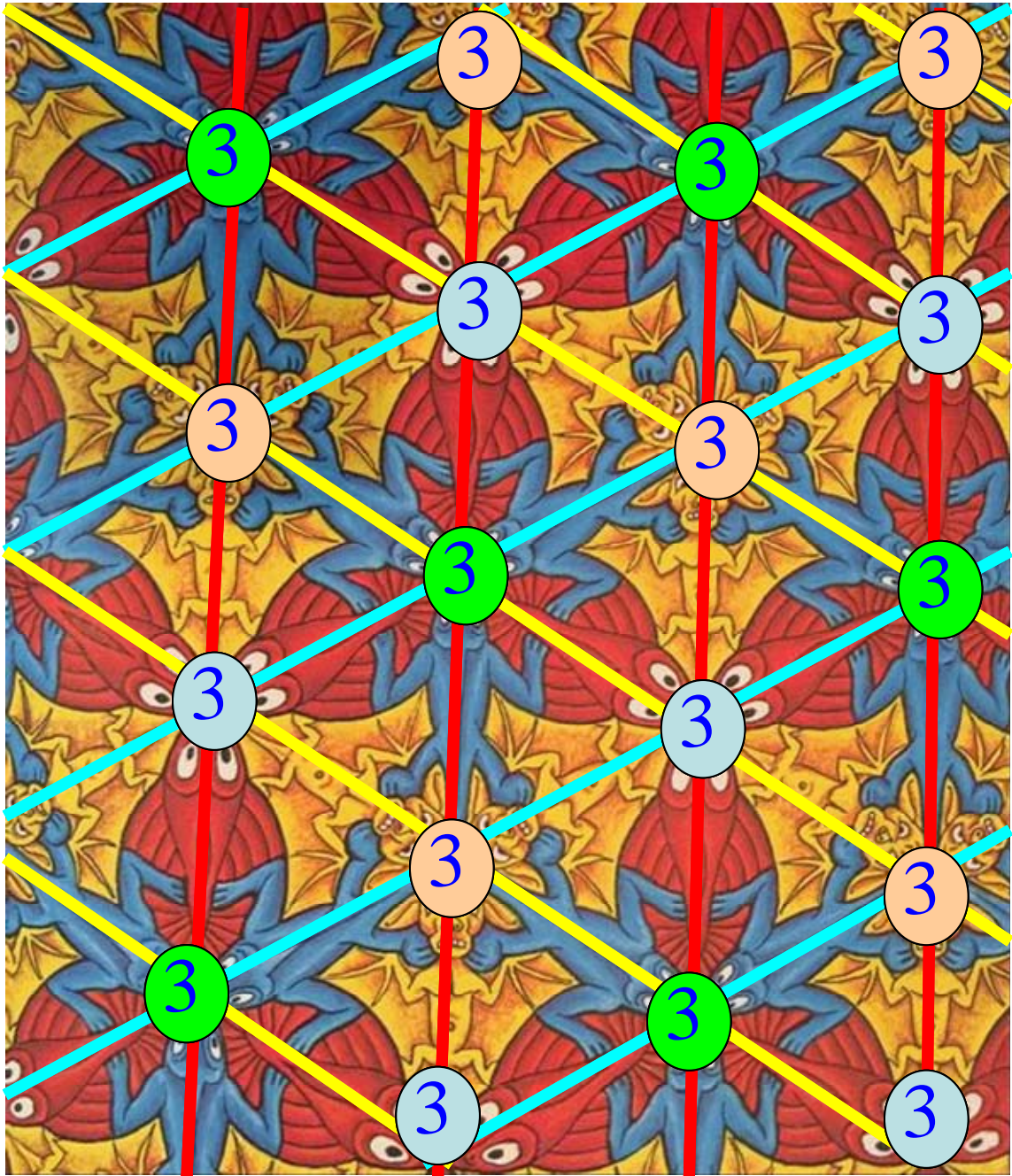
This wallpaper design has no reflections or glide reflections, but it does have rotations of order 3, i.e. rotations through angles of 120 degrees. Below I've marked all the centers of rotation that I've found.



p3m1



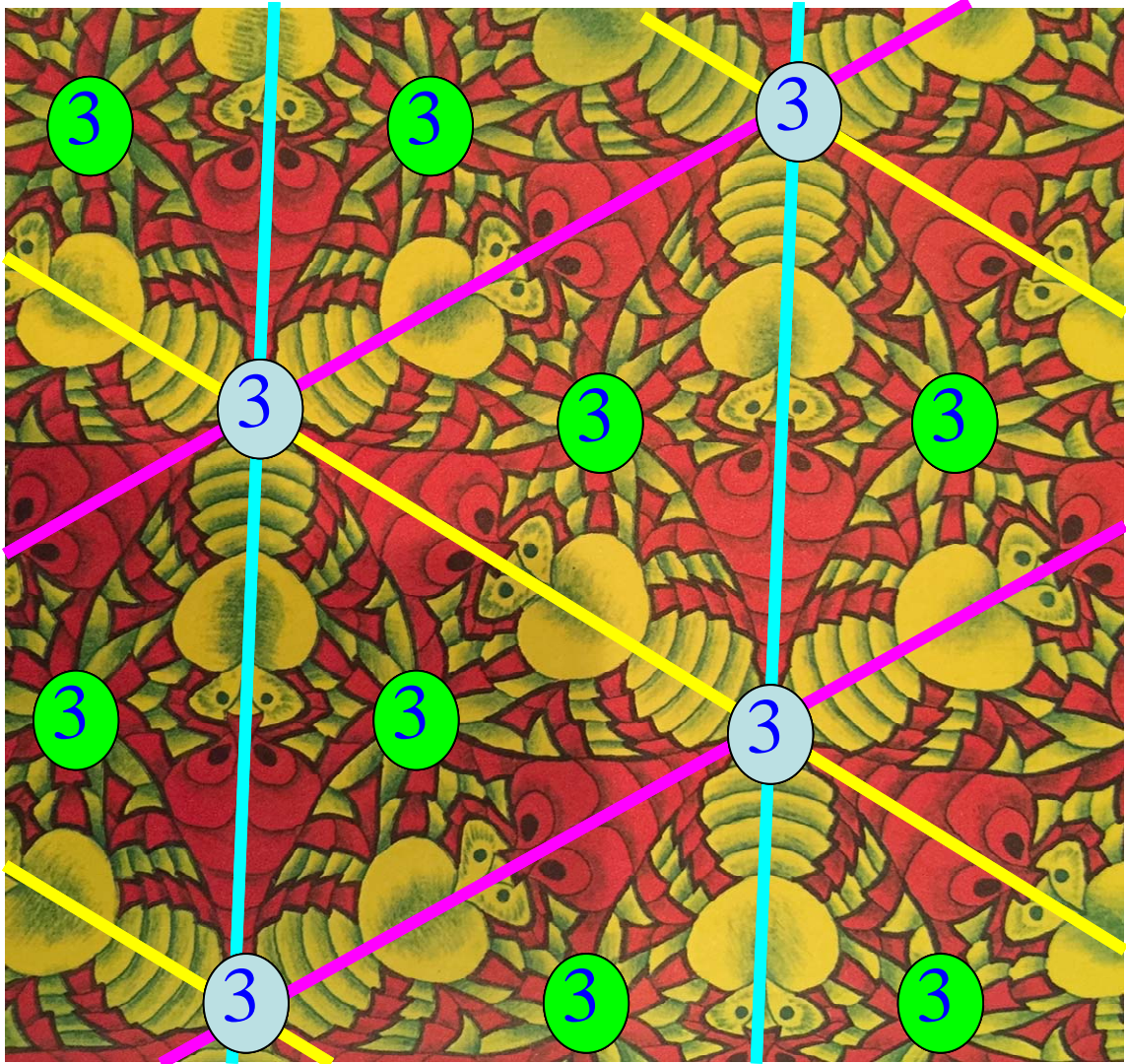
This wallpaper design has rotations of order 3 plus mirror lines, and all the centers of rotation are on the mirror lines. Below is what I found.



p31m

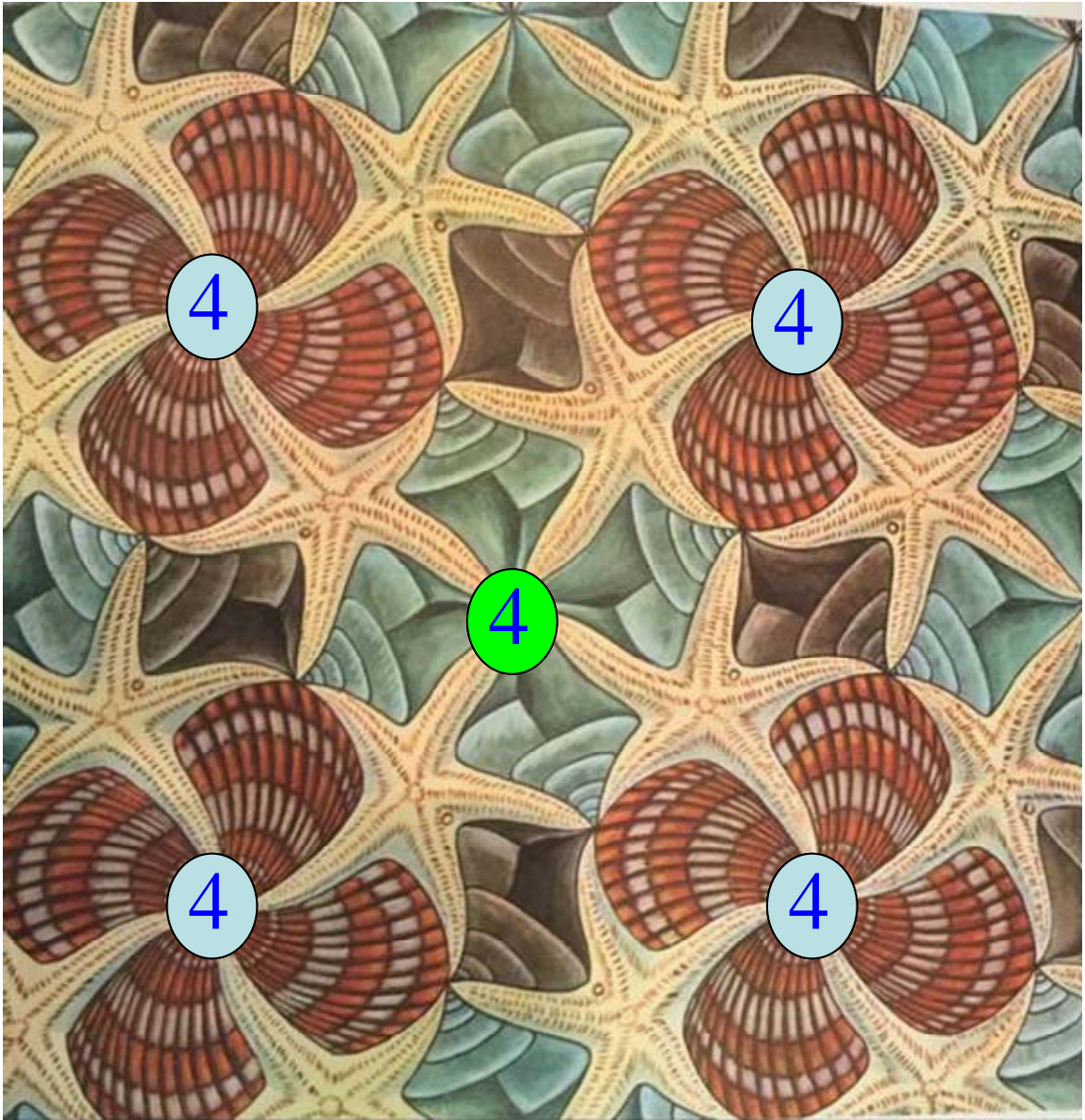


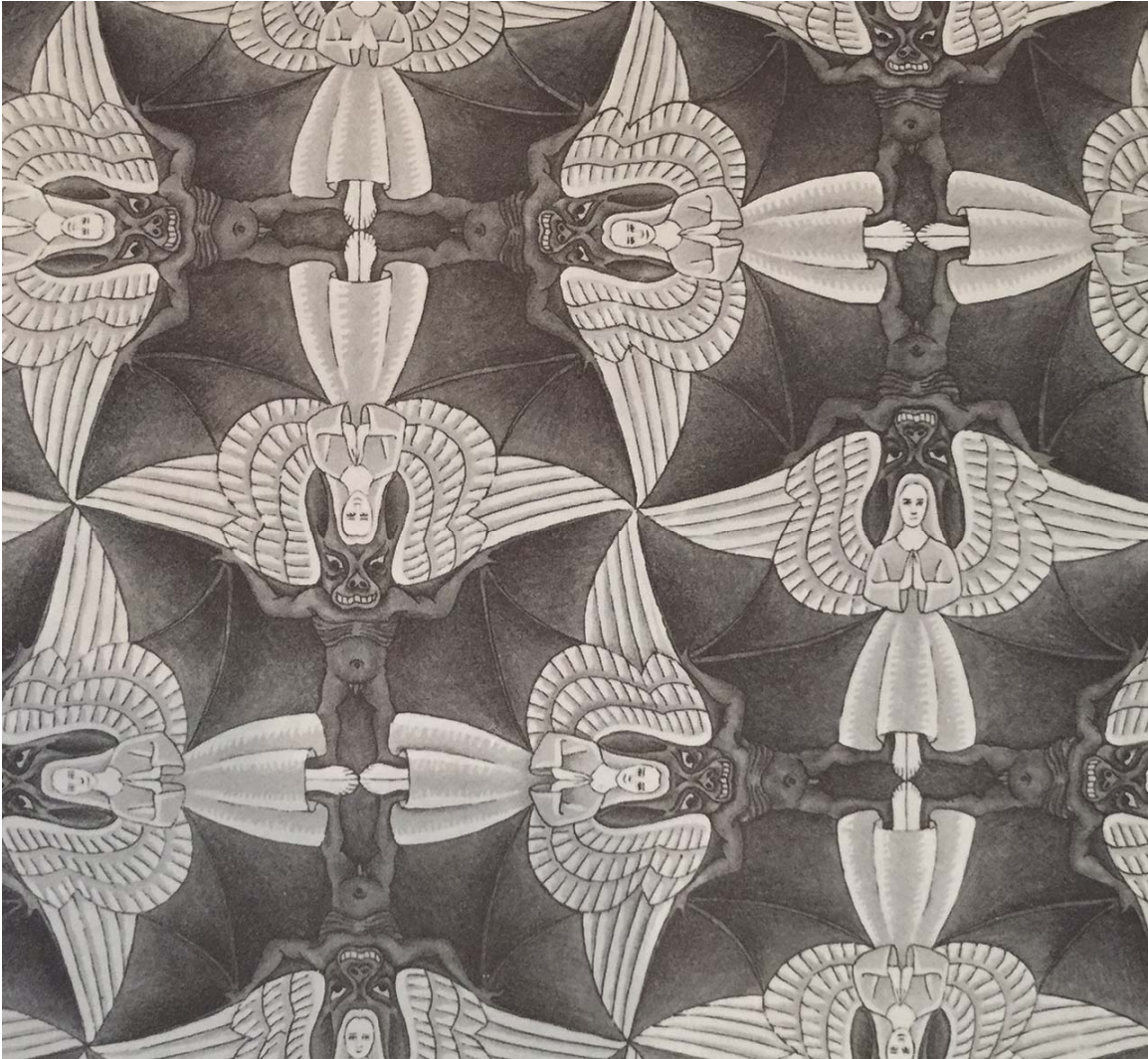
This wallpaper group has reflections and rotations of order 3. However, unlike the previous design, not all of the centers of rotation fall upon mirror lines.



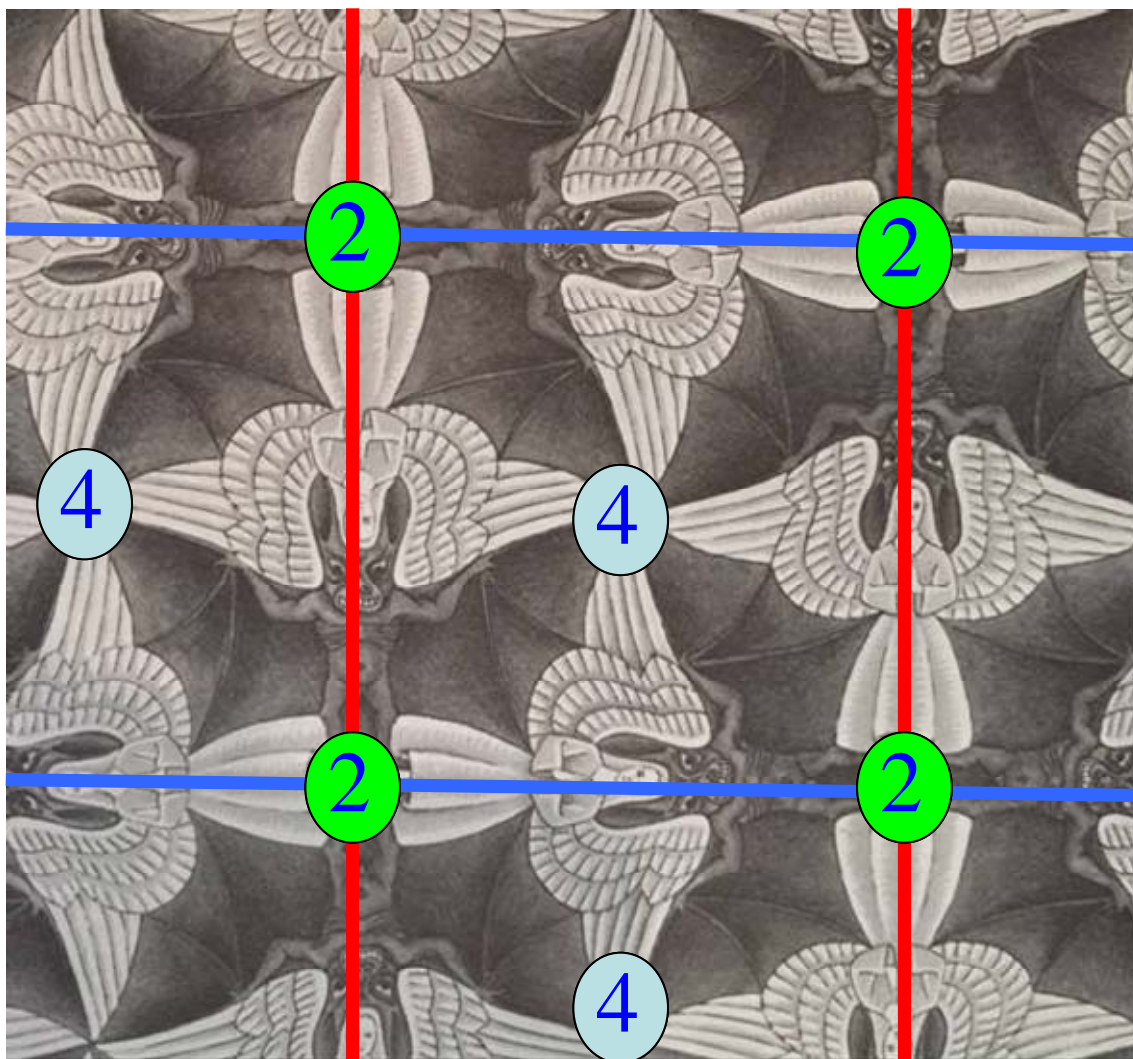


We now move on to wallpaper designs that have rotations of order 4, i.e. 90 degree rotations. This first design contains only rotations, but no reflections or glide reflections.

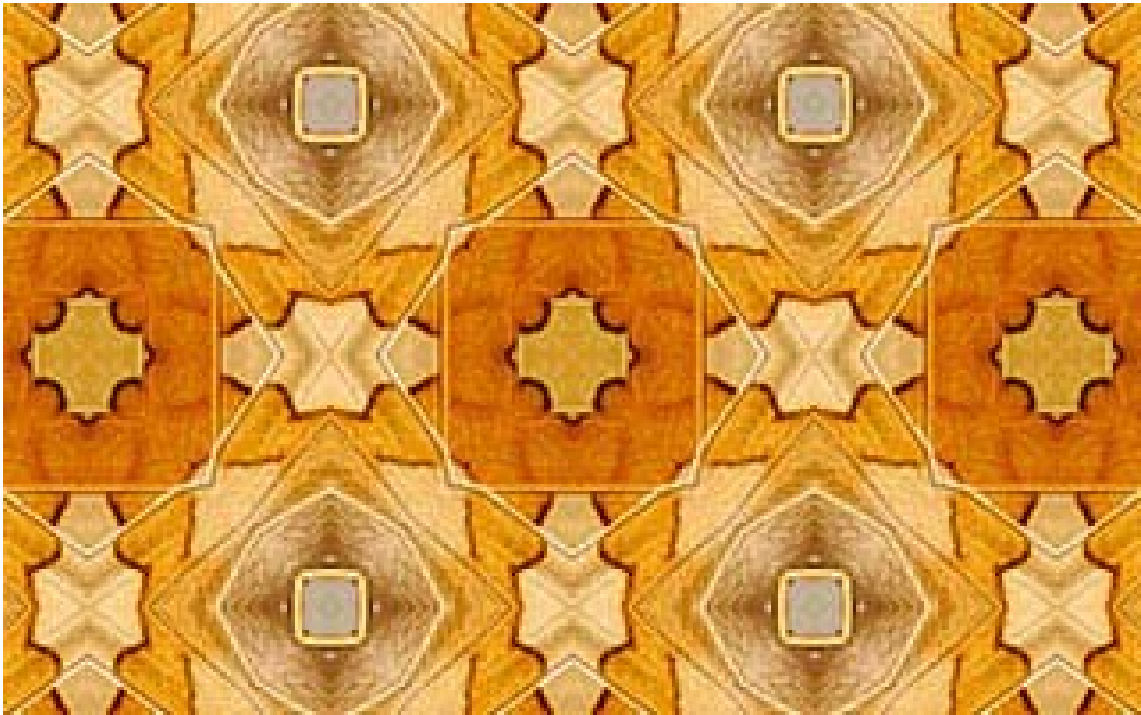




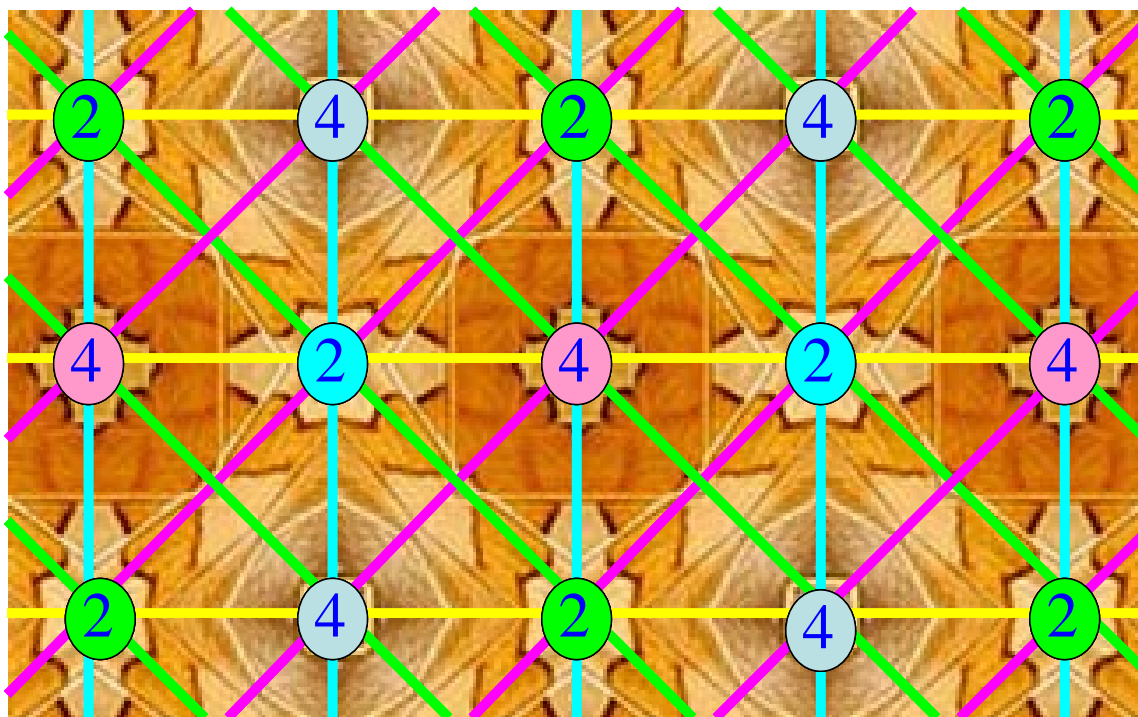
In this design we find rotations of order 4 where the wings of the angels touch, rotations of order 2 where the feet of the angels touch, and mirror lines that do not intersect at 45 degree angles.



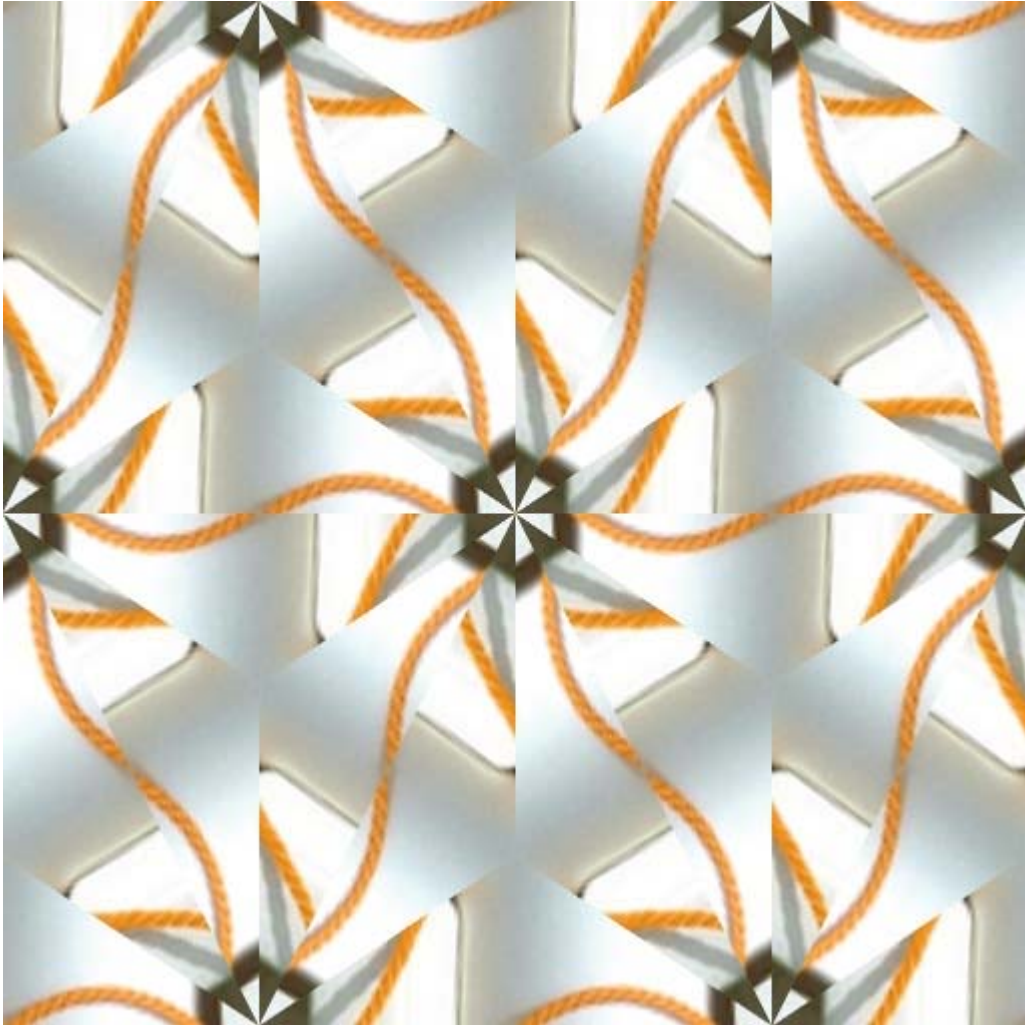
p4m



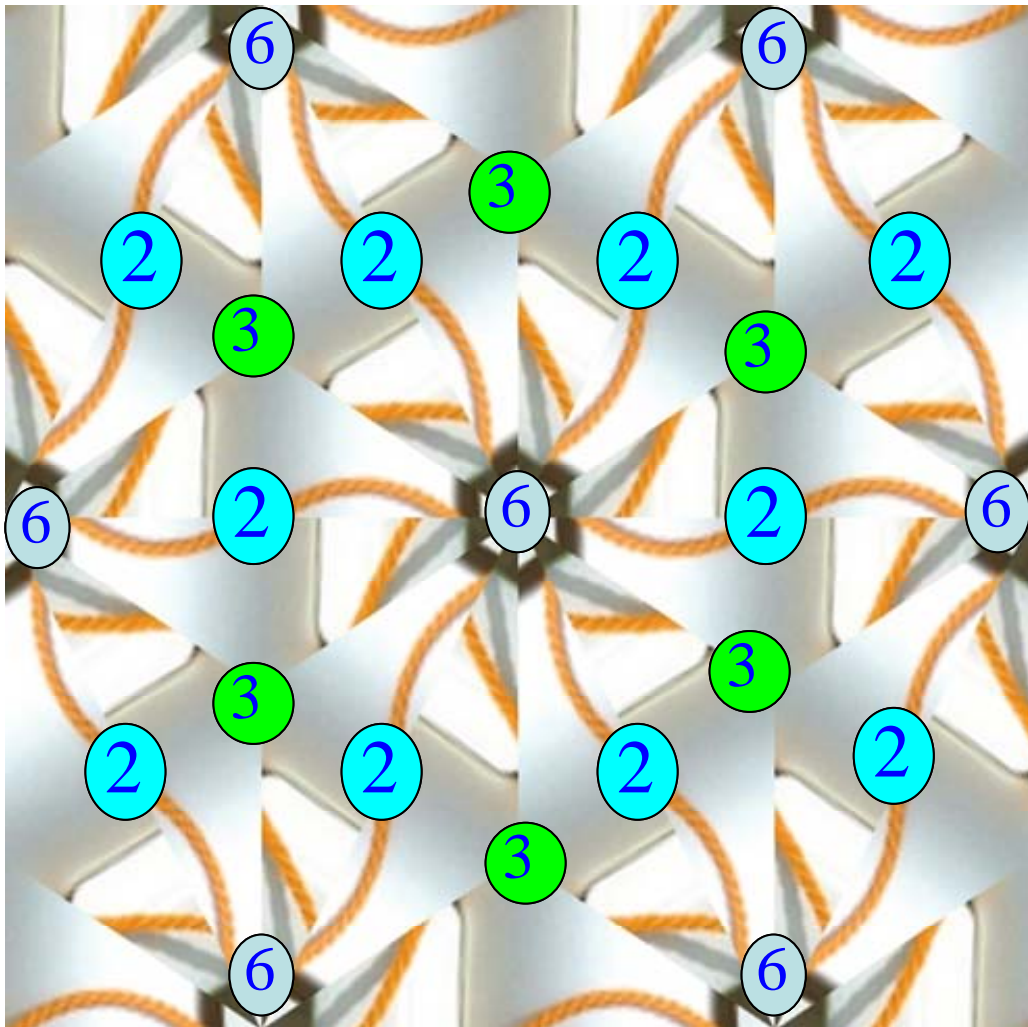
This particular wallpaper design has rotations of order 4, rotations of order 2, and some mirror lines that do intersect at 45 degrees.



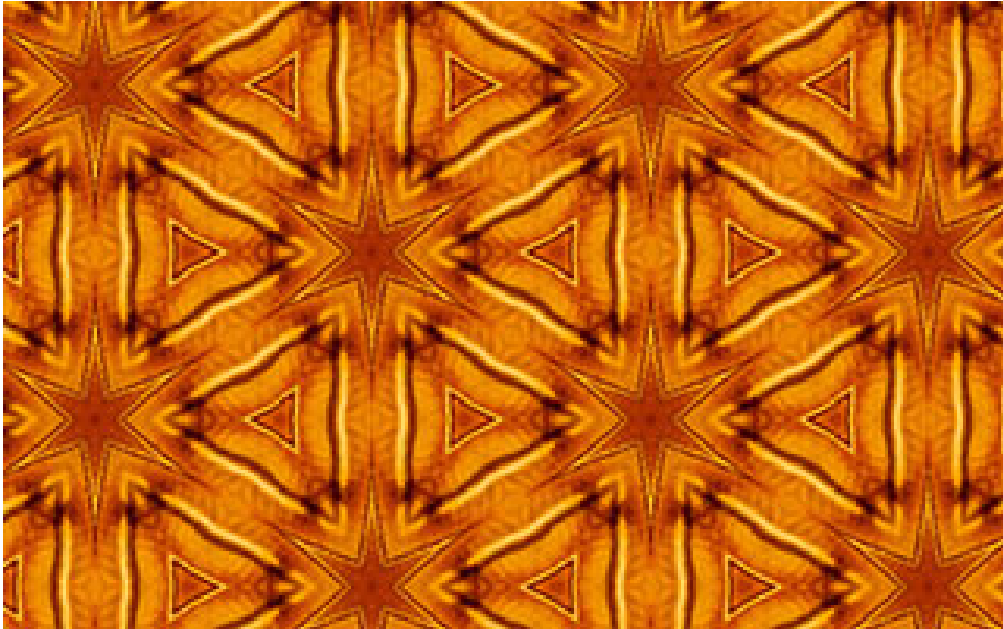
p6



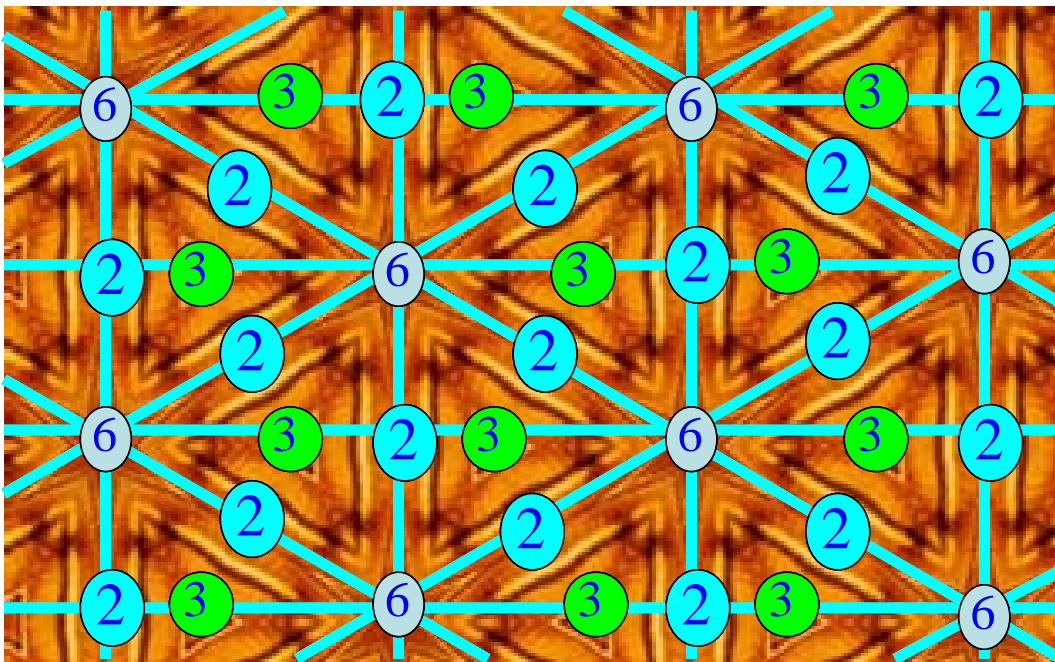
There are only two wallpaper groups left! This one, p6, has rotational symmetry of order 6 (rotations through 60 degree angles), but no reflection or glide reflection lines. It also has some rotations of orders 2 and 3. Once again, below are the centers of rotation that I've found.



p6m



And now we arrive at our final wallpaper group! As with the previous one, this pattern also has rotations of order 6, but it additionally contains several mirror lines.

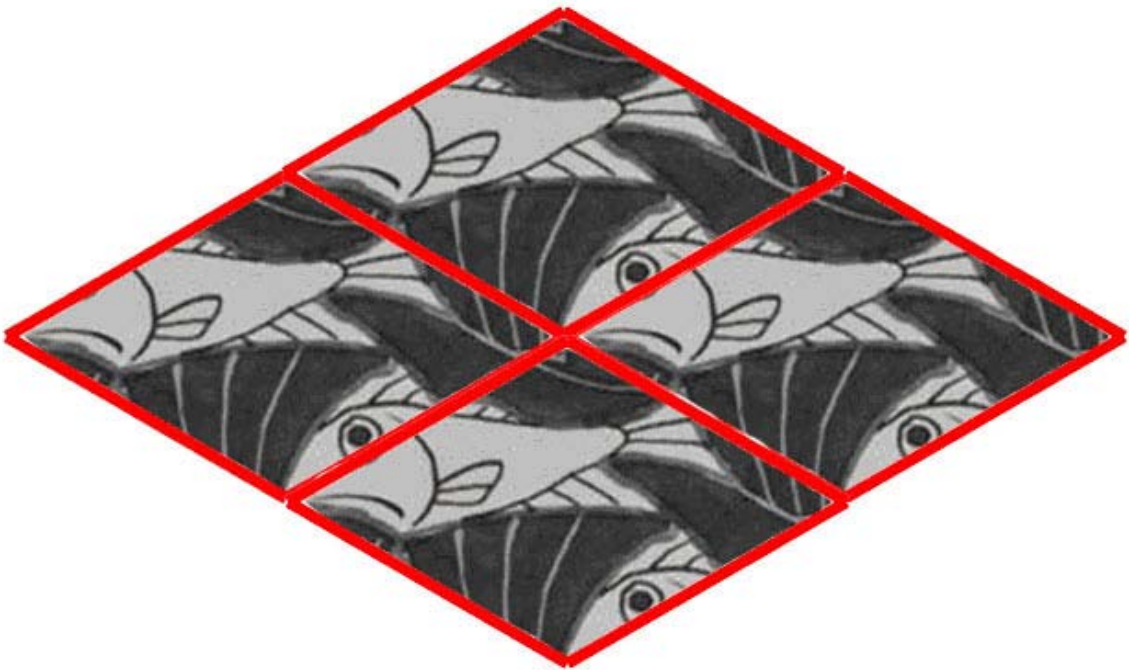
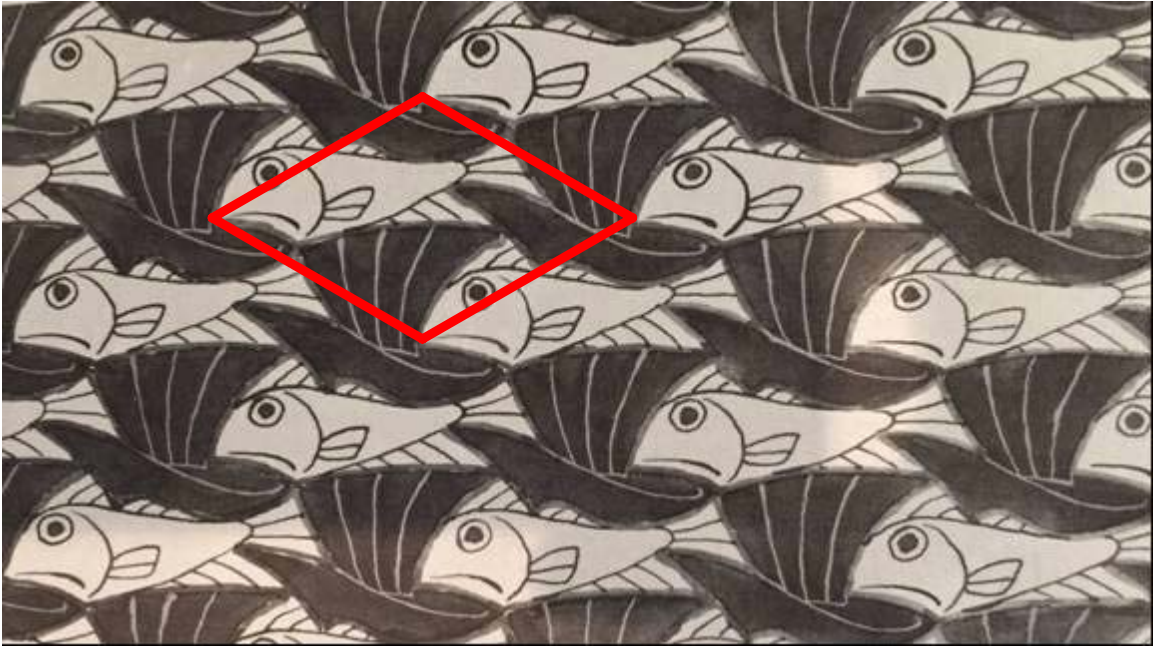


Above I've given you examples of how to classify a wallpaper design by searching out reflections, glide reflections, and centers of rotation. However, one topic I haven't focused on that much (because it wasn't needed for classification) is to find the corresponding lattice and a fundamental region that can generate the entire group through translations, reflections, glide reflections, and rotations. Thus, if you would like to explore this further, below is a repeat of some material I presented earlier, and you can use this to get started. Enjoy!

| Symmetry Group - IUC | Lattice | Rotation | Reflection |
|----------------------|---------------|----------|------------|
| p1 | Parallelogram | none | none |
| p2 | Parallelogram | 2 | none |
| pm | Rectangle | none | parallel |
| pg | Rectangle | none | none |
| cm | Rhombus | none | parallel |
| pmm | Rectangle | 2 | 90 degrees |
| pmg | Rectangle | 2 | parallel |
| pgg | Rectangle | 2 | none |
| cmm | Rhombus | 2 | 90 degrees |
| p4 | Square | 4 | none |
| p4m | Square | 4 + | none |
| p4g | Square | 4 * | 45 degrees |
| p3 | Hexagon | 3 | none |
| p31m | Hexagon | 3 * | 60 degrees |
| p3m1 | Hexagon | 3 + | 30 degrees |
| p6 | Hexagon | 6 | none |
| p6m | Hexagon | 6 | 30 degrees |

The Seventeen Wallpaper Patterns

| Parallelogram | Rectangular | Rhombic | Square | Hexagonal |
|--------------------|---------------------|---------------------|---------------------|----------------------|
| p1 | pm | cm | p4 | p3 |
| p2 | pg | cmm | p4m | p3m1 |
| | pmm | | p4g | p31m |
| | pmg | | | p6 |
| | pgg | | | p6m |



FREE RANGE SYMMETRY

As we've mentioned before, *groups* are related to symmetry, and symmetry is just the repetition of patterns. *Groups* arise from these repetitions when we consider the maneuvers that are required to move a piece of some image from one location to a different one that displays the same pattern. When we consider *frieze groups* and *wallpaper groups*, though, there is one very restrictive element. Namely, that our transformations be rigid ones where the entire plane is being *shifted*, *rotated*, or *reflected* at once. However, I often find this restriction, well, very restrictive, and so I like to indulge in what I call "*free range symmetry*." By this I mean simply look for patterns in your life and your environment, and don't worry about the transformations being rigid. Instead, look for *symmetry* and see how many different kinds of *groups* you might identify. And when you do this, you will gain a greater appreciation for how *groups* and *symmetry* are everywhere in our lives.

Here's an example. Below is a picture from my living room. Let's see what some of the *groups* are that we can identify.



Surrounding the fireplace, we see 18 tiles that are almost identical, and this brings to mind the *cyclic group* of order 18, $C_{18} \cong \mathbb{Z}_{18} \cong \mathbb{Z}_2 \times \mathbb{Z}_9$. We could also construct from these 18 tiles the *symmetric group* of degree 18, i.e. the set of all permutations of 18 objects which yields a group of size $18! \approx 6.402374 \times 10^{15}$. If we look at that tile floor and imagine those rows of square tiles extending to infinity, then the corresponding *group* is *isomorphic* to $\mathbb{Z} \times \mathbb{Z}$. If we now look at the front of the wooden beam above the fireplace, we see a rectangle that can be reflected about either a horizontal axis or a vertical axis. This symmetry results in the *Klein 4-group*, $\mathbb{Z}_2 \times \mathbb{Z}_2$. On top of the wooden beam, we see three copper bowls. As we move from one to another, we see color and shape preserved, but not size. However, if we ignore the change in size and just focus on the symmetry of shape and color, then these three bowls could suggest to us either the *cyclic group* of order 3 or the *symmetric group* of degree 3. Again, *symmetry* doesn't always have to mean repetitions of exact shape. *Symmetry* can result from repetition of colors or general shape or even patterns that we repeat over time. And finally, the guitar to the left of the fireplace exhibits bilateral symmetry. Hence, the *group* $C_2 \cong \mathbb{Z}_2$. Also, the six strings can suggest either the *cyclic group* of order 6, $C_6 \cong \mathbb{Z}_6 \cong \mathbb{Z}_2 \times \mathbb{Z}_3$, or the *symmetric group* of degree 6 that contains $6! = 720$ elements.

Well, these are just a few examples, but hopefully they are enough to make you realize that *symmetry* and *groups* are all around us. And this is no accident. Being able to recognize patterns is a mechanism by which our brain can efficiently organize information with a minimum of effort. For example, if you have a hundred books in your office, then it is much easier for your brain to see all of these objects as examples of just one thing, i.e. books, rather than having to try and juggle a hundred different concepts at once. Thus, *patterns* and *symmetry* help us condense a multitude of information down to a single concept, and that helps our brains more easily cope with what it perceives. Hence, *symmetry* is natural. *Symmetry* is a part of life, and we are designed to perceive

it. And everywhere that there is *symmetry*, there are also *groups*. *Groups* are everywhere. Just go forth and let yourself be aware of that.

HOW TO USE GAP (PART 6)

Once again, there is nothing in this file that was not contained in Part 5. We are just repeating the information for easy reference.

1. *How can I redisplay the previous command in order to edit it?*

Press down on the control key and then also press p. In other words, “Ctrl p”.

2. *If the program gets in a loop and shows you the prompt “brk>” instead of “gap>”, how can I exit the loop?*

Press down on the control key and then also press d. In other words, “Ctrl d”.

3. *How can I exit the program?*

Either click on the “close” box for the window, or type “quit;” and press “Enter.”

4. *How do I find the inverse of a permutation?*

```
gap> a:=(1,2,3,4);  
(1,2,3,4)  
gap> a^-1;  
(1,4,3,2)
```

5. *How can I multiply permutations and raise permutations to powers?*

```
gap> (1,2)*(1,2,3);  
(1,3)
```

```
gap> (1,2,3)^2;  
(1,3,2)
```

```
gap> (1,2,3)^-1;  
(1,3,2)
```

```
gap> (1,2,3)^-2;  
(1,2,3)
```

```
gap> a:=(1,2,3);  
(1,2,3)
```

```
gap> b:=(1,2);  
(1,2)
```

```
gap> a*b;  
(2,3)
```

```
gap> a^2;  
(1,3,2)
```

```
gap> a^-2;  
(1,2,3)
```

```
gap> a^3;  
()
```

```
gap> a^-3;  
()
```

```
gap> (a*b)^2;  
()
```

```
gap> (a*b)^3;  
(2,3)
```

6. *How can I create a group from permutations, find the size of the group, and find the elements in the group?*

```
gap> a:=(1,2);  
(1,2)
```

```
gap> b:=(1,2,3);  
(1,2,3)
```

```
gap> g1:=Group(a,b);  
Group([ (1,2), (1,2,3) ])
```

```
gap> Size(g1);  
6
```

```
gap> Elements(g1);  
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]
```

```
gap> g2:=Group([(1,2),(1,2,3)]);  
Group([ (1,2), (1,2,3) ])
```

```
gap> g3:=Group((1,2),(2,3,4));  
Group([ (1,2), (2,3,4) ])
```

7. *How can I create a cyclic group of order 3?*

```
gap> a:=(1,2,3);  
(1,2,3)
```

```
gap> g1:=Group(a);  
Group([ (1,2,3) ])
```

```
gap> Size(g1);  
3
```

```
gap> Elements(g1);  
[ (), (1,2,3), (1,3,2) ]
```

```
gap> g2:=Group((1,2,3));  
Group([ (1,2,3) ])
```

```
gap> g3:=CyclicGroup(IsPermGroup, 3);  
Group([ (1, 2, 3) ])
```


8. *How can I create a multiplication table for the cyclic group of order 3 that I just created?*

```
gap> ShowMultiplicationTable(g1);
```

```

*      | ()      (1,2,3)  (1,3,2)
-----+-----
()      | ()      (1,2,3)  (1,3,2)
(1,2,3) | (1,2,3) (1,3,2)  ()
(1,3,2) | (1,3,2) ()      (1,2,3)

```

9. *How do I determine if a group is abelian?*

```
gap> g1:=Group((1,2,3));
Group([ (1,2,3) ])
```

```
gap> IsAbelian(g1);
true
```

```
gap> g2:=Group((1,2),(1,2,3));
Group([ (1,2), (1,2,3) ])
```

```
gap> IsAbelian(g2);
false
```

10. *What do I type in order to get help for a command like “Elements?”*

```
gap> ?Elements
```

11. *How do I find all subgroups of a group?*

```
gap> a:=(1, 2, 3);
(1, 2, 3)
```

```

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> Size(g);
6

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> h:=AllSubgroups(g);
[ Group(()), Group([ (2, 3) ]), Group([ (1, 2) ]), Group([ (1, 3) ]),
  Group([ (1, 2, 3) ]), Group([ (1, 2, 3), (2, 3) ]) ]

gap> List(h, i->Elements(i));
[ [ () ], [ (), (2, 3) ], [ (), (1, 2) ], [ (), (1, 3) ], [ (), (1, 2, 3),
  (1, 3, 2) ], [ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ] ]

gap> Elements(h[1]);
[ () ]

gap> Elements(h[2]);
[ (), (2, 3) ]

gap> Elements(h[3]);
[ (), (1, 2) ]

gap> Elements(h[4]);
[ (), (1, 3) ]

gap> Elements(h[5]);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> Elements(h[6]);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

```

12. How do I find the subgroup generated by particular permutations?

```

gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> h:=Subgroup(g, [(1, 2)]);
Group([ (1, 2) ])

gap> Elements(h);
[ (), (1, 2) ]

```

13. How do I determine if a subgroup is normal?

```

gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> h1:=Group((1, 2));
Group([ (1, 2) ])

```

```
gap> IsNormal (g, h1);

gap> h2:=Group((1, 2, 3));
Group([ (1, 2, 3) ])

gap> IsNormal (g, h2);
true
```

14. How do I find all normal subgroups of a group?

```
gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> n:=NormalSubgroups(g);
[ Group([ (1, 2), (1, 2, 3) ]), Group([ (1, 3, 2) ]), Group(()) ]

gap> Elements(n[1]);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> Elements(n[2]);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> Elements(n[3]);
[ () ]
```

15. How do I determine if a group is simple?

```
gap> g:=Group((1, 2), (1, 2, 3));
Group([ (1, 2), (1, 2, 3) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> IsSimple(g);
false

gap> h:=Group((1, 2));
Group([ (1, 2) ])

gap> Elements(h);
[ (), (1, 2) ]

gap> IsSimple(h);
true
```

16. How do I find the right cosets of a subset H of G ?

```
gap> g:=Group([(1, 2, 3), (1, 2)]);
Group([ (1, 2, 3), (1, 2) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> h:=Subgroup(g, [(1, 2)]);
Group([ (1, 2) ])

gap> Elements(h);
[ (), (1, 2) ]

gap> c:=RightCosets(g, h);
[ RightCoset(Group([ (1, 2) ]), ()), RightCoset(Group([ (1, 2) ]), (1, 3, 2)),
  RightCoset(Group([ (1, 2) ]), (1, 2, 3)) ]

gap> List(c, i->Elements(i));
[ [ (), (1, 2) ], [ (2, 3), (1, 3, 2) ], [ (1, 2, 3), (1, 3) ] ]
gap> Elements(c[1]);
[ (), (1, 2) ]

gap> Elements(c[2]);
[ (2, 3), (1, 3, 2) ]

gap> Elements(c[3]);
[ (1, 2, 3), (1, 3) ]

gap> rc:=RightCoset(h, (1, 2, 3));
RightCoset(Group([ (1, 2) ]), (1, 2, 3))

gap> Elements(rc);
[ (1, 2, 3), (1, 3) ]

gap> rc:=h*(1, 2, 3);
RightCoset(Group([ (1, 2) ]), (1, 2, 3))

gap> Elements(rc);
[ (1, 2, 3), (1, 3) ]
```

17. How can I create a quotient (factor) group?

```
gap> g:=Group([(1, 2, 3), (1, 2)]);
Group([ (1, 2, 3), (1, 2) ])

gap> Elements(g);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]

gap> n:=Group((1, 2, 3));
Group([ (1, 2, 3) ])

gap> Elements(n);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> IsNormal(g, n);
true

gap> c:=RightCosets(g, n);
[ RightCoset(Group([ (1, 2, 3) ]), ()), RightCoset(Group([ (1, 2, 3) ]), (2, 3)) ]
```

```

gap> Elements(c[1]);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> Elements(c[2]);
[ (2, 3), (1, 2), (1, 3) ]

gap> f:=FactorGroup(g, n);
Group([ f1 ])

gap> Elements(f);
[ <identity> of ..., f1 ]

gap> ShowMultiplicationTable(f);
*
-----+-----
<identity> of ... | <identity> of ... f1
f1                 | f1                 <identity> of ...

```

18. How do I find the center of a group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> Center(g);
Group(())

gap> c:=Center(g);
Group(())

gap> Elements(c);
[ () ]

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> c:=Center(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(c);
[ (), (1, 3)(2, 4) ]

```

19. How do I find the commutator (derived) subgroup of a group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

```

```

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3, 2) ])

gap> Elements(d);
[ (), (1, 2, 3), (1, 3, 2) ]

gap> a:=(1, 2, 3, 4);
(1, 2, 3, 4)

gap> b:=(1, 3);
(1, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3, 4), (1, 3) ])

gap> d:=DerivedSubgroup(g);
Group([ (1, 3)(2, 4) ])

gap> Elements(d);
[ (), (1, 3)(2, 4) ]

```

20. How do I find all Sylow p -subgroups for a given group?

```

gap> a:=(1, 2, 3);
(1, 2, 3)

gap> b:=(2, 3);
(2, 3)

gap> g:=Group(a, b);
Group([ (1, 2, 3), (2, 3) ])

gap> Size(g);
6

gap> FactorsInt(6);
[ 2, 3 ]

gap> sylow2:=SylowSubgroup(g, 2);
Group([ (2, 3) ])

gap> IsNormal(g, sylow2);
false

gap> c:=ConjugateSubgroups(g, sylow2);
[ Group([ (2, 3) ]), Group([ (1, 3) ]), Group([ (1, 2) ]) ]

gap> Elements(c[1]);
[ (), (2, 3) ]

gap> Elements(c[2]);
[ (), (1, 3) ]

gap> Elements(c[3]);
[ (), (1, 2) ]

gap> sylow3:=SylowSubgroup(g, 3);
Group([ (1, 2, 3) ])

```



```
gap> IsNormal (g, sylow3);
true

gap> Elements(sylow3);
[ (), (1, 2, 3), (1, 3, 2) ]
```

21. How can I create the Rubik's cube group using GAP?

First you need to save the following permutations as a pure text file with the name rubik.txt to your C-drive before you can import it into GAP.

```
r:=(25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24);
l:=(9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35);
u:=(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19);
d:=(41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40);
f:=(17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11);
b:=(33,35,40,38)(34,37,39,36)(3,9,46,32)(2,12,47,29)(1,14,48,27);
```

And now you can read the file into GAP and begin exploring.

```
gap> Read("C:/rubik.txt");

gap> rubik:=Group(r,l,u,d,f,b);
<permutation group with 6 generators>

gap> Size(rubik);
43252003274489856000
```

22. How can I find the center of the Rubik's cube group?

```
gap> c:=Center(rubik);
Group([ (2,34)(4,10)(5,26)(7,18)(12,37)(13,20)(15,44)(21,28)(23,42)(29,36)(31,45)(39,47) ])

gap> Size(c);
2

gap> Elements(c);
[ (), (2,34)(4,10)(5,26)(7,18)(12,37)(13,20)(15,44)(21,28)(23,42)(29,36)(31,45)(39,47) ]
```



```
gap> Elements(sylow11);
[ (), (4, 5, 36, 21, 31, 15, 39, 13, 42, 7, 12) (10, 26, 29, 28, 45, 44, 47, 20, 23, 18, 37),
(4, 7, 13, 15, 21, 5, 12, 42, 39, 31, 36) (10, 18, 20, 44, 28, 26, 37, 23, 47, 45, 29),
(4, 12, 7, 42, 13, 39, 15, 31, 21, 36, 5) (10, 37, 18, 23, 20, 47, 44, 45, 28, 29, 26),
(4, 13, 21, 12, 39, 36, 7, 15, 5, 42, 31) (10, 20, 28, 37, 47, 29, 18, 44, 26, 23, 45),
(4, 15, 12, 31, 7, 21, 42, 36, 13, 5, 39) (10, 44, 37, 45, 18, 28, 23, 29, 20, 26, 47),
(4, 21, 39, 7, 5, 31, 13, 12, 36, 15, 42) (10, 28, 47, 18, 26, 45, 20, 37, 29, 44, 23),
(4, 31, 42, 5, 15, 7, 36, 39, 12, 21, 13) (10, 45, 23, 26, 44, 18, 29, 47, 37, 28, 20),
(4, 36, 31, 39, 42, 12, 5, 21, 15, 13, 7) (10, 29, 45, 47, 23, 37, 26, 28, 44, 20, 18),
(4, 39, 5, 13, 36, 42, 21, 7, 31, 12, 15) (10, 47, 26, 20, 29, 23, 28, 18, 45, 37, 44),
(4, 42, 15, 36, 12, 13, 31, 5, 7, 39, 21) (10, 23, 44, 29, 37, 20, 45, 26, 18, 47, 28) ]
```

```
gap> IsNormal(rubik, sylow2);
false
```

```
gap> IsNormal(rubik, sylow3);
false
```

```
gap> IsNormal(rubik, sylow5);
false
```

```
gap> IsNormal(rubik, sylow7);
false
```

```
gap> IsNormal(rubik, sylow11);
false
```

NOTE: All of the *Sylow p -subgroups* found above have *conjugates*, but the sheer size of the *Rubik's cube group* makes it too difficult to pursue them on a typical desktop computer.

26. How do I determine if a group is cyclic?

```
gap> a:=(1, 2, 3)*(4, 5, 6, 7);
(1, 2, 3)(4, 5, 6, 7)
gap> g:=Group(a);
Group([ (1, 2, 3)(4, 5, 6, 7) ])
gap> Size(g);
12
gap> IsCyclic(g);
true
```

27. How do I create a dihedral group with $2n$ elements for an n -sided regular polygon?

```
gap> d4:=DihedralGroup(IsPermGroup, 8);
Group([ (1, 2, 3, 4), (2, 4) ])
```

```
gap> Elements(d4);
[ (), (2, 4), (1, 2)(3, 4), (1, 2, 3, 4), (1, 3), (1, 3)(2, 4), (1, 4, 3, 2), (1, 4)(2, 3) ]
```

28. How can I express the elements of a dihedral group as rotations and flips rather than as permutations?

```
gap> d3:=DihedralGroup(6);
<pc group of size 6 with 2 generators>

gap> Elements(d3);
[ <identity> of ..., f1, f2, f1*f2, f2^2, f1*f2^2 ]

gap> ShowMultiplicationTable(d3);
*-----|<identity> of ... f1      f2      f1*f2      f2^2      f1*f2^2 \
<identity> of ... |<identity> of ... f1      f2      f1*f2      f2^2      f1*f2^2
f1                |f1                <identity> of ... f1*f2      f2      f1*f2^2      f2^2
f2                |f2                f1*f2^2      f2^2      f1      <identity> of ... f1*f2
f1*f2             |f1*f2             f2^2      f1*f2^2      <identity> of ... f1
f2^2              |f2^2              f1*f2      f1*f2^2      f2      <identity> of ... f2
f1*f2^2           |f1*f2^2           f2      f1      f1*f2^2      f2      <identity> of ... f1
```

29. How do I create a symmetric group of degree n with $n!$ elements?

```
gap> s4:=SymmetricGroup(4);
Sym( [ 1 .. 4 ] )

gap> Size(s4);
24

gap> Elements(s4);
[ (), (3, 4), (2, 3), (2, 3, 4), (2, 4, 3), (2, 4), (1, 2), (1, 2)(3, 4), (1, 2, 3),
(1, 2, 3, 4), (1, 2, 4, 3), (1, 2, 4), (1, 3, 2),
(1, 3, 4, 2), (1, 3), (1, 3, 4), (1, 3)(2, 4), (1, 3, 2, 4), (1, 4, 3, 2), (1, 4, 2), (1, 4, 3),
(1, 4), (1, 4, 2, 3), (1, 4)(2, 3) ]
```

30. How do I create an alternating group of degree n with $\frac{n!}{2}$ elements?

```
gap> a4:=AlternatingGroup(4);
Alt( [ 1 .. 4 ] )

gap> Size(a4);
12

gap> Elements(a4);
[ (), (2, 3, 4), (2, 4, 3), (1, 2)(3, 4), (1, 2, 3), (1, 2, 4), (1, 3, 2), (1, 3, 4),
(1, 3)(2, 4), (1, 4, 2), (1, 4, 3), (1, 4)(2, 3) ]
```

31. How do I create a direct product of two or more groups?

```
gap> g1:=Group((1, 2, 3));
Group([ (1, 2, 3) ])
```



```

gap> g2:=Group((4,5));
Group([ (4,5) ])

gap> dp:=DirectProduct(g1,g2);
Group([ (1,2,3), (4,5) ])

gap> Size(dp);
6
gap> Elements(dp);
[ (), (4,5), (1,2,3), (1,2,3)(4,5), (1,3,2), (1,3,2)(4,5) ]

gap> ShowMultiplicationTable(dp);
*
(1,3,2)(4,5) | ()          (4,5)          (1,2,3)          (1,2,3)(4,5) (1,3,2)
-----
---
()            | ()          (4,5)          (1,2,3)          (1,2,3)(4,5) (1,3,2)
(1,3,2)(4,5) | (4,5)        ()          (1,2,3)(4,5) (1,2,3) (1,3,2)(4,5) (1,3,2)
(4,5)         | (1,2,3)      (1,2,3)(4,5) (1,3,2) (1,3,2)(4,5) () (4,5)
(1,2,3)       | (1,2,3)(4,5) (1,2,3) (1,3,2)(4,5) (1,3,2) (4,5)
(1,2,3)(4,5) | (1,3,2)      (1,3,2)(4,5) ()          (4,5) (1,2,3)
(1,2,3)       | (1,3,2)(4,5) (1,3,2) (4,5)          ()          (1,2,3)(4,5) (1,2,3)
(1,3,2)(4,5) | (1,3,2)(4,5) (1,3,2) (4,5)          ()          (1,2,3)(4,5) (1,2,3)

```

32. How can I create the Quaternion group?

```

gap> a:=(1,2,5,6)*(3,8,7,4);
(1,2,5,6)(3,8,7,4)

gap> b:=(1,4,5,8)*(2,7,6,3);
(1,4,5,8)(2,7,6,3)

gap> q:=Group(a,b);
Group([ (1,2,5,6)(3,8,7,4), (1,4,5,8)(2,7,6,3) ])

gap> Size(q);
8

gap> IsAbelian(q);
false

gap> Elements(q);
[ (), (1,2,5,6)(3,8,7,4), (1,3,5,7)(2,4,6,8), (1,4,5,8)(2,7,6,3),
  (1,5)(2,6)(3,7)(4,8), (1,6,5,2)(3,4,7,8),
  (1,7,5,3)(2,8,6,4), (1,8,5,4)(2,3,6,7) ]

gap> q:=QuaternionGroup(IsPermGroup,8);
Group([ (1,5,3,7)(2,8,4,6), (1,2,3,4)(5,6,7,8) ])

gap> Size(q);
8

gap> IsAbelian(q);
false

gap> Elements(q);
[ (), (1,2,3,4)(5,6,7,8), (1,3)(2,4)(5,7)(6,8), (1,4,3,2)(5,8,7,6),
  (1,5,3,7)(2,8,4,6), (1,6,3,8)(2,5,4,7),
  (1,7,3,5)(2,6,4,8), (1,8,3,6)(2,7,4,5) ]

```

33. How can I find a set of independent generators for a group?

```
gap> c6:=CyclicGroup(IsPermGroup, 6);  
Group([ (1, 2, 3, 4, 5, 6) ])
```

```
gap> Size(c6);  
6
```

```
gap> GeneratorsOfGroup(c6);  
[ (1, 2, 3, 4, 5, 6) ]
```

```
gap> d4:=DihedralGroup(IsPermGroup, 8);  
Group([ (1, 2, 3, 4), (2, 4) ])
```

```
gap> Size(d4);  
8
```

```
gap> GeneratorsOfGroup(d4);  
[ (1, 2, 3, 4), (2, 4) ]
```

```
gap> s5:=SymmetricGroup(5);  
Sym([ 1 .. 5 ])
```

```
gap> Size(s5);  
120
```

```
gap> GeneratorsOfGroup(s5);  
[ (1, 2, 3, 4, 5), (1, 2) ]
```

```
gap> a5:=AlternatingGroup(5);  
Alt([ 1 .. 5 ])
```

```
gap> Size(a5);  
60
```

```
gap> GeneratorsOfGroup(a5);  
[ (1, 2, 3, 4, 5), (3, 4, 5) ]
```

```
gap> q:=QuaternionGroup(IsPermGroup, 8);  
Group([ (1, 5, 3, 7)(2, 8, 4, 6), (1, 2, 3, 4)(5, 6, 7, 8) ])
```

```
gap> Size(q);  
8
```

```
gap> GeneratorsOfGroup(q);  
[ (1, 5, 3, 7)(2, 8, 4, 6), (1, 2, 3, 4)(5, 6, 7, 8) ]
```

34. How do I find the conjugate of a permutation in the form $a^b = b^{-1}ab$?

```
gap> a:=(1, 2, 3, 4, 5);  
(1, 2, 3, 4, 5)
```

```
gap> b:=(2, 4, 5);  
(2, 4, 5)
```

```
gap> a^b;
(1, 4, 3, 5, 2)
```

```
gap> b^-1*a*b;
(1, 4, 3, 5, 2)
```

35. *How do I divide up a group into classes of elements that are conjugate to one another? (Note that “conjugacy” is an equivalence relation on our group G . That means that G can be separated into nonintersecting subsets that contain only elements that are conjugate to one another.)*

```
gap> d3:=DihedralGroup(IsPermGroup, 6);
Group([ (1, 2, 3), (2, 3) ])
```

```
gap> Size(d3);
6
```

```
gap> Elements(d3);
[ (), (2, 3), (1, 2), (1, 2, 3), (1, 3, 2), (1, 3) ]
```

```
gap> cc:=ConjugacyClasses(d3);
[ ()^G, (2, 3)^G, (1, 2, 3)^G ]
```

```
gap> Elements(cc[1]);
[ () ]
```

```
gap> Elements(cc[2]);
[ (2, 3), (1, 2), (1, 3) ]
```

```
gap> Elements(cc[3]);
[ (1, 2, 3), (1, 3, 2) ]
```

SUMMARY (PART 6)

In this part we continued our examination of visual representations of *groups*. In particular, you should be familiar with the following, and you should develop the habit of identifying *groups*, *symmetry*, and *cycles* in your life, both in space and in time, whenever they occur!

- *Frieze Groups.*
- *Wallpaper Groups.*
- *Free Range Symmetry.*

PRACTICE (PART 6)

FRIEZE GROUPS

Use the flow chart from the lesson to classify each frieze pattern below as either F1, F2, F3, F4, F5, F6, or F7.

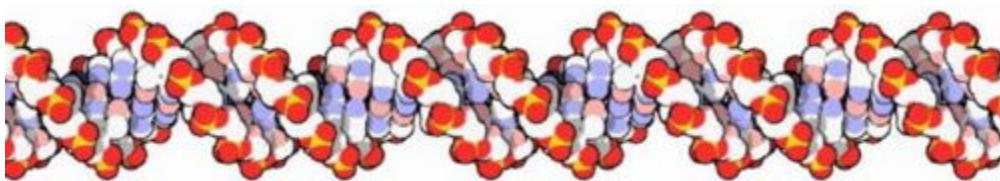
1.



2.



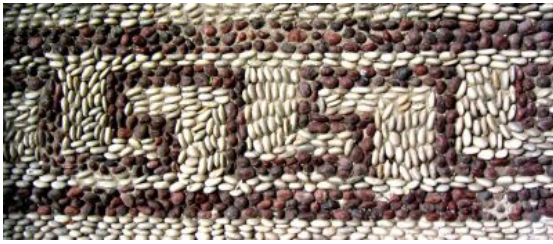
3.



4.



5.



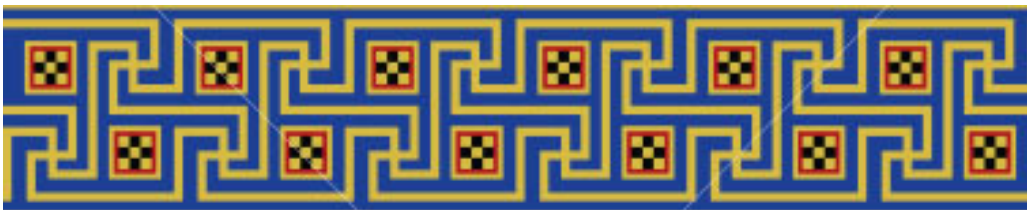
6.



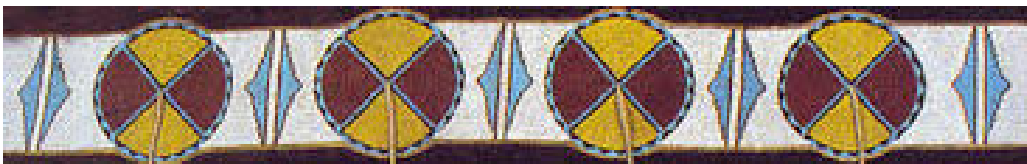
7.



8.



9.

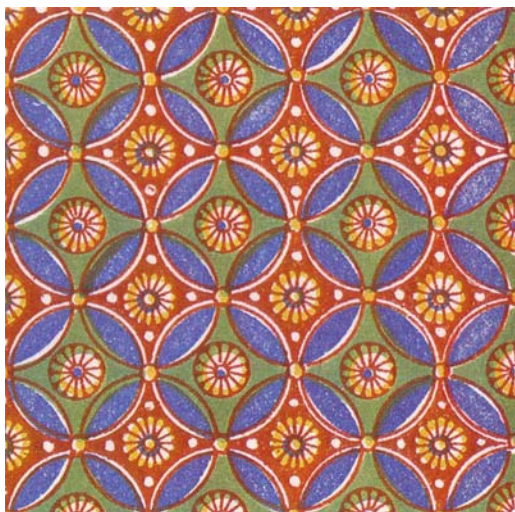


10. Either through drawing by hand or by using a program such as Photoshop, create examples of each of the seven *frieze groups*.

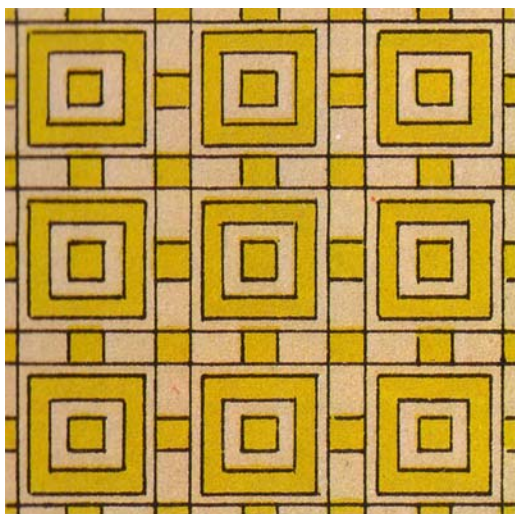
WALLPAPER GROUPS

Use the flowchart from the lesson to identify the *wallpaper group* for each image below.

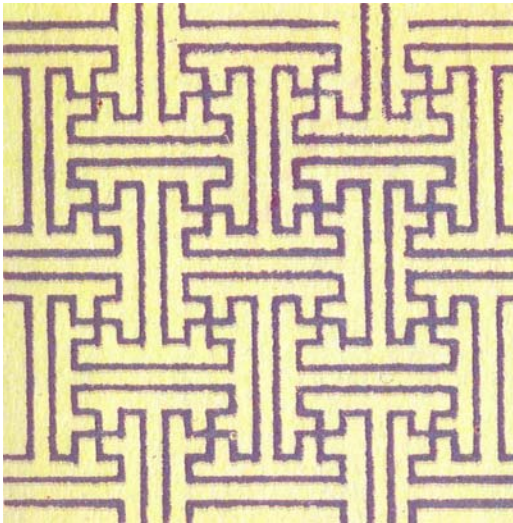
1.



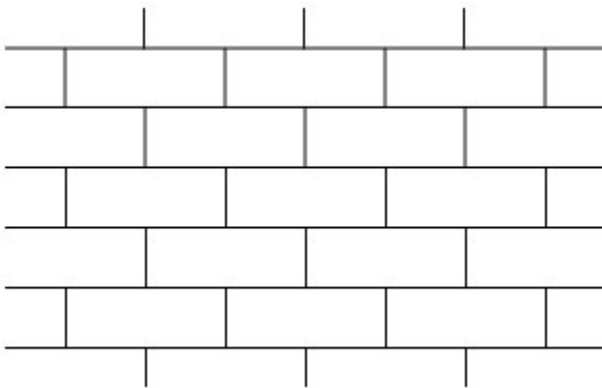
2.



3.



4.



5.



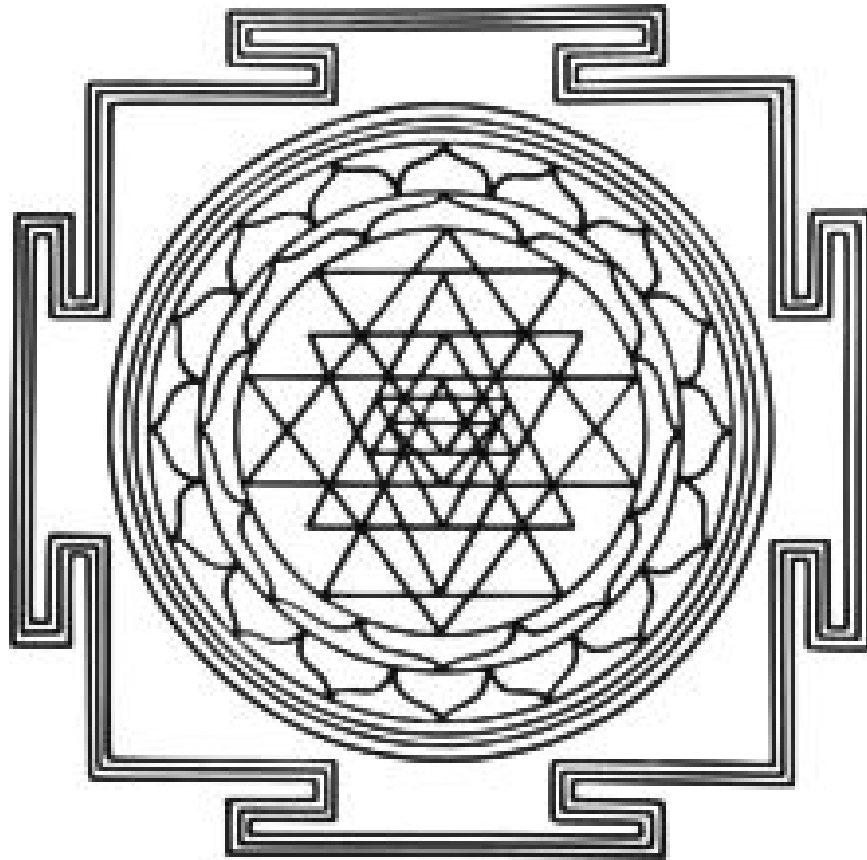
FREE RANGE SYMMETRY

Now that you have some understanding about how *groups* and *symmetry* are everywhere, make a list of some of the *symmetries* and *groups* that you encounter in your life and in your environment. And don't restrict yourself to just completely identical, congruent shapes. Look for repetition of color, repetition of shapes that change size, and cycles that occur over time. Look at your repetitive habits and behaviors as well, and consider ways in which you can combine *cycles* in order to produce more complex *groups*. By doing this, you will not only develop a greater appreciation for how ubiquitous *groups* and *symmetry* are in your life, you may also arrive at a greater understanding of the internal *cycles* and *patterns* that define the person that you are. *Groups*, *symmetry*, and *cycles* are everywhere. Learn to see this!

"The sun also rises, and the sun goes down, and hastens to its place where it rises again. The wind goes toward the south, and turns around to the north; it whirls around continually, and the wind returns again according to its circuits. All the rivers run into the sea; yet the sea is not full; to the place from where the rivers come, there they return again. (Ecclesiastes 1:5-7)

And finally, examine the two images below and identify as many *groups* as you can. Also, don't restrict yourself to just *cyclic groups*. Imagine cycles that overlap with one another in ways that create *nonabelian groups*! For example, in each eye socket of the *día de los muertos* skull there is a flower with five red petals. Imagine labeling the petals one through five and then consider the *group* generated by the cycles $(1,2,3)$ and $(3,4,5)$!





PRACTICE (PART 6) - ANSWERS

FRIEZE GROUPS

Use the flow chart from the lesson to classify each frieze pattern below as either F1, F2, F3, F4, F5, F6, or F7.

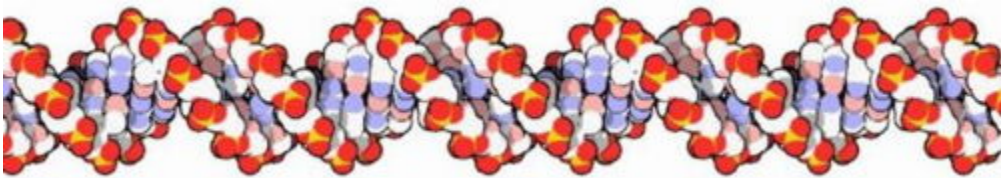
1. F6 – Jump



2. F3 – Sidle



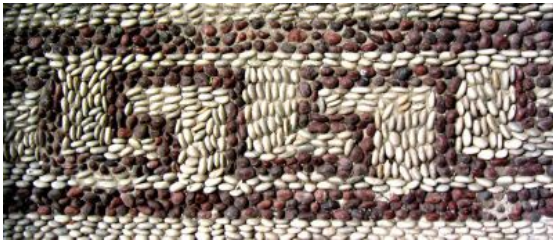
3. F4 – Spinning Sidle



4. F2 – Step



5. F4 – Spinning Sidle



6. F1 – Hop



7. F1 – Hop



8. F4 – Spinning Hop



9. F7 – Spinning Jump

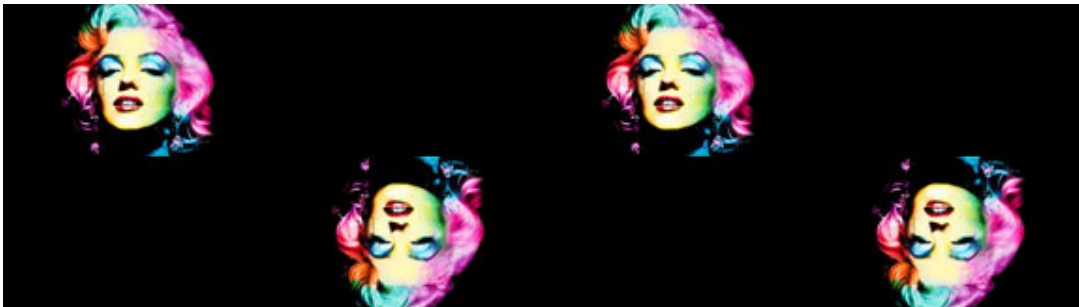


10. Either through drawing by hand or by using a program such as Photoshop, create examples of each of the seven frieze groups.

F1 – Hop



F2 – Step



F3 – Sidle



F4 – Spinning Hop



F5 – Spinning Sidle



F6 – Jump



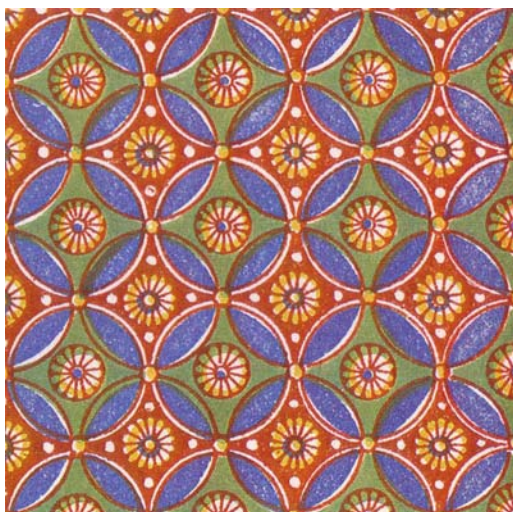
F7 – Spinning Jump



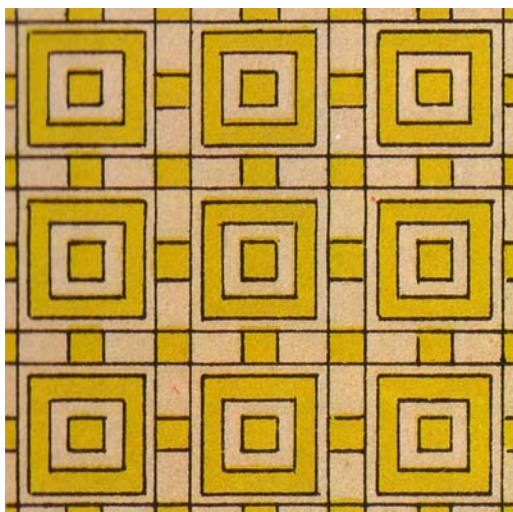
WALLPAPER GROUPS

Use the flowchart in this lesson to identify the *wallpaper group* for each image below.

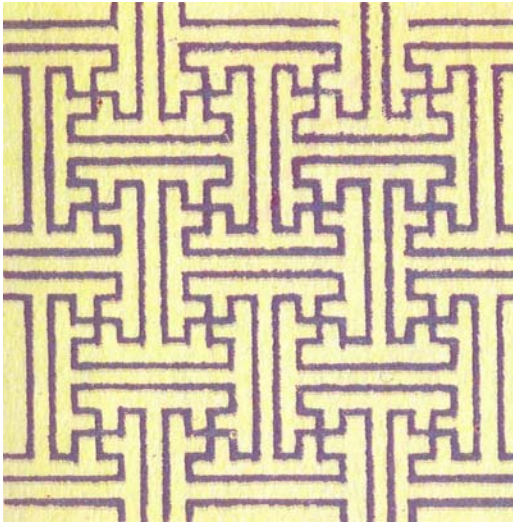
1. p4m



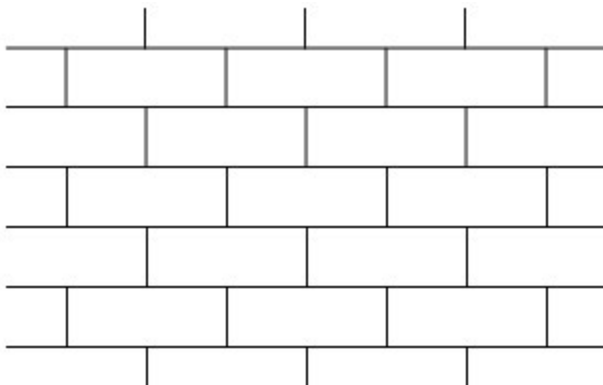
2. p4m



3. p4g



4. cmm



5. p2



FREE RANGE SYMMETRY

Now that you have some understanding about how *groups* and *symmetry* are everywhere, make a list of some of the *symmetries* and *groups* that you encounter in your life and in your environment. And don't restrict yourself to just completely identical, congruent shapes. Look for repetition of color, repetition of shapes that change size, and cycles that occur over time. Look at your repetitive habits and behaviors as well, and consider ways in which you can combine *cycles* in order to produce more complex *groups*. By doing this, you will not only develop a greater appreciation for how ubiquitous *groups* and *symmetry* are in your life, you may also arrive at a greater understanding of the internal *cycles* and *patterns* that define the person that you are. *Groups*, *symmetry*, and *cycles* are everywhere. Learn to see this!

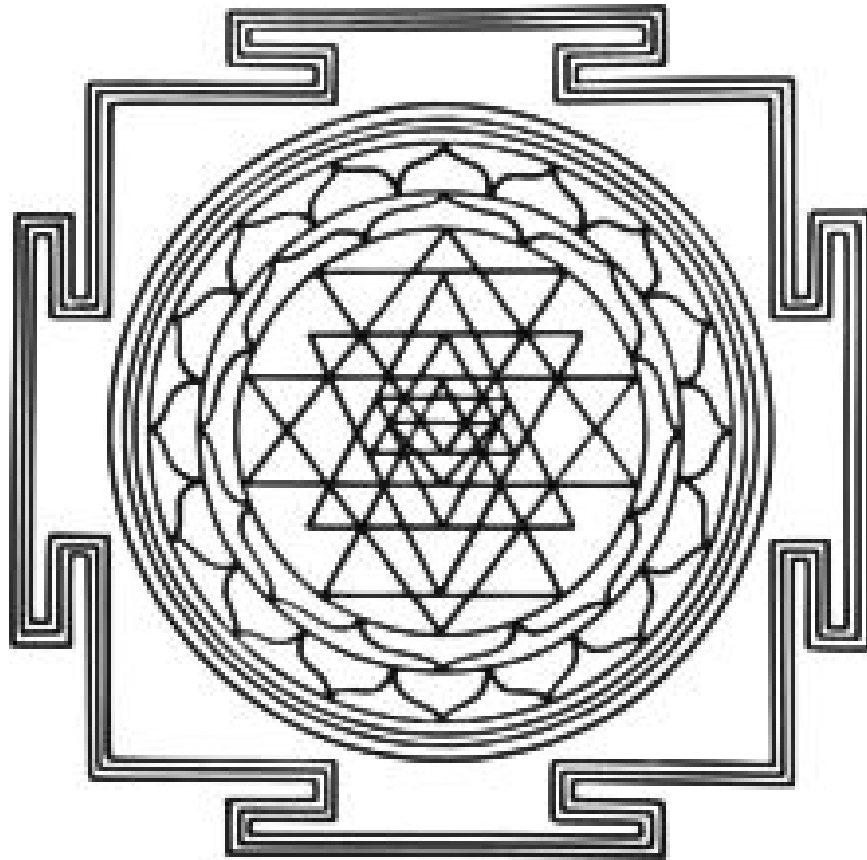
"The sun also rises, and the sun goes down, and hastens to its place where it rises again. The wind goes toward the south, and turns around to the north; it whirls around continually, and the wind returns again according to its circuits. All the rivers run into the sea; yet the sea is not full; to the place from where the rivers come, there they return again. (Ecclesiastes 1:5-7)

And finally, examine the two images below and identify as many *groups* as you can. Also, don't restrict yourself to just *cyclic groups*. Imagine cycles that overlap with one another in ways that create *nonabelian groups*! For example, in each eye socket of the *día de los muertos* skull there is a flower with five red petals. Imagine labeling the petals one through five and then consider the *group* generated by the cycles $(1,2,3)$ and $(3,4,5)$!

(Your answers will vary!)



Día de Los Muertos Skull



Sri Yantra



**YOU CAN LEARN GROUP THEORY IN A
FLASH!**