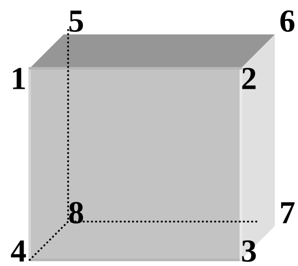
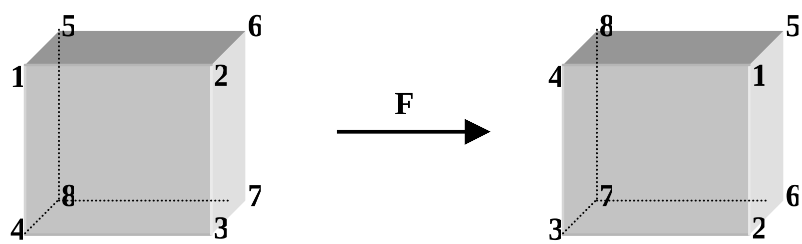
An Introduction to GAP

In today's lesson we will be introduced to **gap**, a freely distributed computer program designed to handle large computations within and relating to groups. Your previous experience with the computer algebra system *Maple* will prove helpful because gap was modeled after Maple. Although **gap** lacks Maple's graphical capabilities and fancy interface, the syntax of the two languages are very similar. Like *Maple*, **gap** has an extensive online help facility. In fact, all of gap's 800+ page manual is available via key-word search from within the program. It is also worth noting that **gap** is created (the development is ongoing) and used by researchers; the program is very powerful in allowing mathematicians to explore open questions in computational group theory. Let's begin our exploration with an example.

Consider the set of rigid motions on a cube. We describe them as permutations on the set of labels {1, 2, 3, 4, 5, 6, 7, 8} which enumerate the corners of the cube:



For example, a clockwise rotation of the cube about an axis through the center of the front and back faces would be described as follows:



F = (1, 2, 3, 4)(5, 6, 7, 8)

The rotation F will generate a group of order 4 since four clockwise turns of the front face brings the cube back to its original position.

We pursue this example with **gap**. Since there are 8 corners to the cube, we are working within the symmetric group on 8 elements. To define G to be the symmetric group on 8 elements (i.e., S_8) we use the following command:

```
gap>  G:=SymmetricGroup(8);
```

Note the use of ":=" in the statement above. This syntax is identical to that of *Maple*. In particular, **[name] := [object];** is always used for an *assignment* statement.

Like *Maple*, every statement must end with a semi-colon. If the semi-colon were not there, **gap** would wait. When you hit the <return> key, **gap** responds on the next line with a value for the statement you entered. Try this now. Then enter the following line to define the rotation F as the permutation (1, 2, 3, 4)(5, 6, 7 ,8).

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

The rotation F will generate a cyclic subgroup of order four:

```
gap>  H1:=Subgroup(G,[F]);
gap>  Elements(H1);
gap>  Size(H1);
```

Next look at the subgroup generated by two types of rotations. (Which rotation does R represent?)

```
gap>  R:=(1, 5, 8, 4)(2, 6, 7, 3);
gap>  H2:=Subgroup(G,[F,R]);
gap>  Elements(H2);
gap>  Size(H2);
```

With the help of **gap** you can find out much more about the groups H1 and H2…

Are either of them abelian?

```
gap>  IsAbelian(H1);
gap>  IsAbelian(H2);
```

What are the orders of elements in each group?

```
gap>  List(Elements(H1), x-> Order(H1,x));
gap>  List(Elements(H2), x-> Order(H2,x));
```

Which elements in the group have order 3?

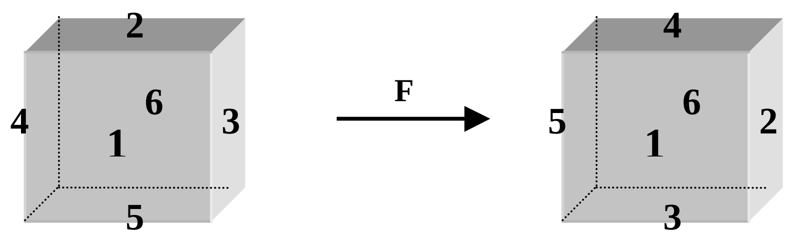
```
gap>  Filtered(Elements(H1), x-> Order(H1,x)=3);
gap>  Filtered(Elements(H2), x-> Order(H2,x)=3);
```

The possibilities of the program are endless!

Question 1. Use **gap** to compute the product R*F. What is the order of R*F? Explain what this motion is geometrically.

Question 2. Use **gap** to compute the product (F*R)^2. What is the order of (F*R)^2? Explain what this rotation is geometrically.

Question 3. Another way to represent the symmetries of the cube is to enumerate the 6 faces.

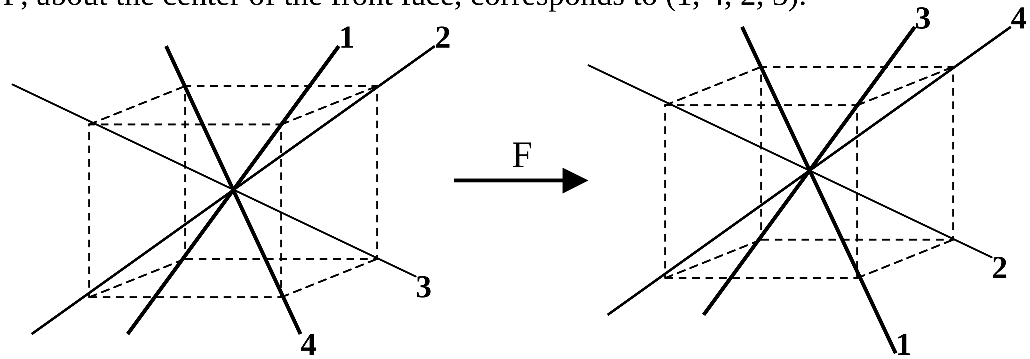


Under this description, the group of rotations on the cube is a subgroup of S_6.
We start over with G defined to be S_6.

```
>gap  quit;
>G:=SymmetricGroup(6);
```

a.) How would you represent F and R under this new labeling? Define F and R (in **gap**) to be the appropriate permutations in S_6 .
b.) Compute R*F and (F*R)^2. How do your results compare to what you got when you enumerated the corners of the cube?

Question 4. A third way to represent the symmetries of the cube is to enumerate the 4 long diagonals of the cube. For example, if we label them as shown below the rotation, F, about the center of the front face, corresponds to (1, 4, 2, 3).



a.) Using this representation, how would you describe R? Use **gap** to compute the products R*F and (F*R)^2. Remember, you will need to redefine these elements first.
b.) Give a geometrical description of each of these products (that is, describe the axis of rotation and the magnitude of the rotation.)