FUN WITH MULTIPLICATION TABLES

Look at the following multiplication table for the set $M = \{m1, m2\}$:

*	m1	m2
m1	m1	m2
m2	m2	m1

Recall that for M to be a group, it needs to have an associative binary operations, an identity, and every element needs to have an inverse. Since M is given by a multiplication table, you can check that the binary operation is well defined by looking to make sure that everything in the table is in M.

1.1 Check to see whether M is a group. Write out each equation that you have to check to see that the operation on M is associative, that M has an identity, and that each element of M has an inverse.

Next we will have GAP do the same thing. You will have to read in the functions, Assoc, Ident, Inver. You can do this with the `Read` command. For example;

```
gap>Read("Assoc");
```

Once you have read in all of these functions, you can run once you put your table in. You do that with the command `Magma` [1] Use the command:

```
gap>m:=MagmaByMultiplicationTable([[1,2],[2,1]]);
```

Notice that when we do this, we put in the 1 and the 2 from the first line of the table, but not the m. GAP will put the m in for you. Unfortunately, it seems to be the only letter it will put in, so we should call all of our sets with a binary operation m. So it goes.

[1]**Magma** is not a standard mathematical term. It means a set with a binary operation. It was used by a famous fictional French mathematician (and real French general).

Next run the functions you read in, so that you can check your calculations from 1.1. Start with Assoc.

```
gap>Assoc(m);
```

This will give you each equation you should have checked to see if M was associative, along with whether the equation is true or false. Next check if each element is an identity with Ident.

```
gap>Ident(m)
```

This will check to see if each element of M is an identity. Last, check if each element has an inverse with Inver.

```
gap>Inver(m,e);
```

Here e is the **integer** corresponding to the identity you found. For example if the identity was $m2$ you would type `Inver(m,2);` .

1.2 You have now checked all the group properties. Is M a group? After you have an answer, check it by using the command `AsGroup(m);` .

1.3 Now do the same thing with the following multiplication table. Check and see if it is a group by hand, then check using `Assoc, Inver,` and `Ident` . Only use `AsGroup` after you have checked everything else.

*	m1	m2	m3
m1	m1	m2	m2
m2	m1	m2	m3
m3	m1	m3	m2

You can also check sets we have already seen, like S_3 and D_4 using these functions.

1.4 Write out the multiplication table for S_3. Would you like to check by hand that this is a group.

1.5 Check S_3 using the functions we used above. To get S_3 you need the command

```
gap>s3 := SymmetricGroup(3);
```