14 Multiplication Modulo m Meets Groups

1. **Clearly some of these numbers cannot be elements of a group. For instance, in both cases, 0 cannot be used, since it prevents the existence of an inverse. In the case of mod 4, 2 cannot be used either. Why not?**

2 can't be used because it lacks an inverse. After all, there is no integer x such that $2x \equiv 1 (\mathrm{mod}\, 4)$.

2. **How could we have known that these numbers would not work in advance?**

They share a common factor with the modulus m; we must have $\gcd(x, m) = 1$ where x is the element of the group.

3. **Euler's totient function $\varphi(m)$ tells us how many numbers are relatively prime to a given number m. That is, $\varphi(m)$ is the count of numbers n such that $\gcd(m, n) = 1$. What does the maximum size of a group under multiplication mod m have to do with this function?**

The size of the largest group under multiplication modulo m is $\varphi(m)$, since that's the group of numbers which have inverses modulo m. The other group properties are satisfied; associativity because multiplication is associative, closure because the product of two numbers coprime to m is also coprime to m, and identity, since 1 is always (considered) coprime to m.

4. **We will write tables for the largest possible groups under multiplication mod 5 and mod 8.**

   (a) **Make a prediction as to how many elements will be in each group.**

(Answers may vary.)
Since 5 is prime, there should be 4 elements in its group since $\varphi(p)$ with p prime is $p - 1$. For 8, the elements should be $1, 3, 5, 7$: the odd numbers, or 4 elements.

   (b) **Which numbers can you eliminate from consideration?**

We can eliminate numbers that are coprime to the modulus. For 5, this is $\{0\}$. For 8, this is $\{0, 2, 4, 6\}$.

   (c) **Do you think that the groups will be isomorphic to those of multiplication mod 3 and mod 4, or to each other?**

(Answers may vary.)
The group mod 5, since 5 is prime, should just be the cyclic group of order 4. 8 is harder to predict; it turns out to be a different group of order 4. Since multiplication groups mod $3, 4$ have order 2, it can't be isomorphic to these.

   (d) **Find the period of each element in the groups and write their orbits: the list of its powers until it reaches the identity.**

mod 5:

Element	Orbit	Period
1	$1, (1)$	1
2	$2, 4, 3, 1, (2)$	4
3	$3, 4, 2, 1, (3)$	4
4	$4, 1, (4)$	2

mod 8:

Element	Orbit	Period
1	$1, (1)$	1
3	$3, 1, (3)$	3
5	$5, 1, (5)$	5
7	$7, 1, (7)$	7

   (e) **Make the tables, and analyze them to confirm/correct your predictions.**

The multiplicative group of integers mod 5 is indeed the cyclic group of order 4, since there are elements of periods $1, 2$, and 4. If we want to write a correspondence, let the cyclic group of order 4 be $\{I, r, r^2, r^3\}$ under multiplication. Then $1 \leftrightarrow I$, $2 \leftrightarrow r$, $3 \leftrightarrow r^3$, $4 \leftrightarrow r^2$.
The multiplicative group of integers mod 8 is the dihedral group of order 4, or D_2.

   (f) **Are there any subgroups?**

Yes; $\{1, 4\}$ is a subgroup of the mod 5 multiplicative group, and $\{1, 3\}$, $\{1, 5\}$, $\{1, 7\}$ are subgroups of the mod 8 multiplicative group. $\{1\}$ is a subgroup of both groups.

5. Now use a program to find the largest possible group under multiplication mod 14.

Here's my Python code for this, compressed to fit:

```
#!/usr/bin/env python
from fractions import gcd

# run in a terminal as python -i mod_m_find.py,
# so you can interact with it as a REPL

modulus = 10
group_elements = []
for i in xrange(1, modulus):
    if gcd(i, modulus) == 1:
        group_elements.append(i)
def pretty_print_elements():
    print(pp_list(group_elements))
def pp_list(arr):
    return ", ".join(map(lambda s: "$%s$" % s, arr))
def make_orbit_table(sort_by_orbit=True):
    print("\\begin{tabular}{c|l|c}\nElement & Orbit & Period \\\\ \\hline")
    orbits = []
    for elem in group_elements:
        x = elem
        orbit = []
        while True:
            orbit.append(str(x))
            x *= elem
            x %= modulus

            if x == elem:
                orbit.append("(%s)" % elem)
                break
        orbits.append(orbit)
    if sort_by_orbit:
        orbits.sort(key=len)
    for orbiit in orbits:
        print("$%s$ & %s & $%s$ \\\\" % (orbiit[0], pp_list(orbiit), len(orbiit) - 1))
    print("\\end{tabular}")
def create_group_table(sort_by=None):
    if not sort_by:
        group_elements.sort()
    else:
        group_elements.sort(key=sort_by)
    hline = "\\hline"
    print("\\begin{array}{c|%s}" % ("c|" * len(group_elements)))
    print("\\cdot & " + " & ".join(map(str, group_elements)) + " \\\\ " + hline)
    for i in group_elements:
        row = "%s" % i
        for j in group_elements:
            row += " & %s" % ((i * j) % modulus)
        row += " \\\\ " + hline
        print(row)
    print("\\end{array}")
```

You can find this code at *https://github.com/anematode/gatm/blob/master/accessories/scripts/mod_m_find.py*.

(a) What are its elements?

```
Timothys-Pro:gatm timoothy$ python -i accessories/scripts/mod_m_find.py
```

```
>>> pretty_print_elements()
$1$, $3$, $7$, $9$
```

My program is customized for LaTeX of course, since that's what I'm writing this text in. But the elements are $\{1, 3, 7, 9\}$.

(b) Make a table of the group's orbits.

```
>>> make_orbit_table(False)
\begin{tabular}{c|l|c}
Element & Orbit & Period \\
$1$ & $1$, $(1)$ & $1$ \\
$3$ & $3$, $9$, $7$, $1$, $(3)$ & $4$ \\
$7$ & $7$, $9$, $3$, $1$, $(7)$ & $4$ \\
$9$ & $9$, $1$, $(9)$ & $2$ \\
\end{tabular}
```

This renders to the following table of orbits.

Element	Orbit	Period
1	1, (1)	1
3	3, 9, 7, 1, (3)	4
7	7, 9, 3, 1, (7)	4
9	9, 1, (9)	2

(c) Make a group table.

```
>>> create_group_table()
\begin{array}{c|c|c|c|c|}
\cdot & 1 & 3 & 7 & 9 \\ \hline
1 & 1 & 3 & 7 & 9 \\ \hline
3 & 3 & 9 & 1 & 7 \\ \hline
7 & 7 & 1 & 9 & 3 \\ \hline
9 & 9 & 7 & 3 & 1 \\ \hline
\end{array}
```

This gives the following table:

$\cdot$	1	3	7	9
1	1	3	7	9
3	3	9	1	7
7	7	1	9	3
9	9	7	3	1

(d) It might be good to order the numbers at the top of the table so that they start with a 1 and go by successive powers of 3.

The successive powers of 3 are $1, 3, 9, 7$.

```
>>> create_group_table([1,3,9,7].index)
\begin{array}{c|c|c|c|c|}
\cdot & 1 & 3 & 9 & 7 \\ \hline
1 & 1 & 3 & 9 & 7 \\ \hline
3 & 3 & 9 & 7 & 1 \\ \hline
9 & 9 & 7 & 1 & 3 \\ \hline
7 & 7 & 1 & 3 & 9 \\ \hline
\end{array}
```

This gives the following table:

$\cdot$	1	3	9	7
1	1	3	9	7
3	3	9	7	1
9	9	7	1	3
7	7	1	3	9

(e) What group is it isomorphic to?

It is now clear that this is isomorphic to the cyclic group of order 4, C_4.

(f) Does it have any subgroups; if so, what are they?

It does have subgroups: the trivial subgroup $\{1\}$ and the cyclic group of order 2, $\{1, 9\}$.

6. Now, a surprise: find the powers of 10, mod 14.

Here's my one-liner in Python to do this:

```
>>> print(";\\quad ".join("10^%s \\equiv %s (\\operatorname{mod} 14)" %
  (n, 10**n % 14) for n in range(1,8)))
```

$$10^1 \equiv 10 (\mathrm{mod}\, 14); \quad 10^2 \equiv 2 (\mathrm{mod}\, 14); \quad 10^3 \equiv 6 (\mathrm{mod}\, 14); \quad 10^4 \equiv 4 (\mathrm{mod}\, 14);$$

$$10^5 \equiv 12 (\mathrm{mod}\, 14); \quad 10^6 \equiv 8 (\mathrm{mod}\, 14); \quad 10^7 \equiv 10 (\mathrm{mod}\, 14).$$

It cycles!

(a) How long is the period of this orbit?

The orbit appears to be 6 elements long, since $7 - 1 = 6$.

(b) What number appears to be the identity element?

$10^6 = 8$ appears to be the identity element, since multiplying it by 10 yields 10.

(c) Make a table in which the identity element comes first.

```
>>> elements = [8,10,2,6,4,12];print("\\cdot & " + " & ".join(
  map(str,elements)) + " \\\\ \\hline");print("\n".join(
    " & ".join([str(i)] + map(lambda x: str((x * i) % 14), elements))
\cdot & 8 & 10 & 2 & 6 & 4 & 12 \\ \hline
8 & 8 & 10 & 2 & 6 & 4 & 12 \\ \hline
10 & 10 & 2 & 6 & 4 & 12 & 8 \\ \hline
2 & 2 & 6 & 4 & 12 & 8 & 10 \\ \hline
6 & 6 & 4 & 12 & 8 & 10 & 2 \\ \hline
4 & 4 & 12 & 8 & 10 & 2 & 6 \\ \hline
12 & 12 & 8 & 10 & 2 & 6 & 4 \\ \hline
```

Here's how it looks:

$\cdot$	8	10	2	6	4	12
8	8	10	2	6	4	12
10	10	2	6	4	12	8
2	2	6	4	12	8	10
6	6	4	12	8	10	2
4	4	12	8	10	2	6
12	12	8	10	2	6	4

(d) Find a number besides 10 whose group of powers mod 14 is isomorphic to this group.

This is clearly the cyclic group of order 6, so we can choose any generator of this group to get the same group. We might choose 12 for example.

```
>>> print(";\\quad ".join("12^%s \\equiv %s (\\operatorname{mod} 14)" %
  (n, 12**n % 14) for n in range(1,8)))
```

$$12^1 \equiv 12 (\mathrm{mod}\, 14); \quad 12^2 \equiv 4 (\mathrm{mod}\, 14); \quad 12^3 \equiv 6 (\mathrm{mod}\, 14); \quad 12^4 \equiv 2 (\mathrm{mod}\, 14);$$
$$12^5 \equiv 10 (\mathrm{mod}\, 14); \quad 12^6 \equiv 8 (\mathrm{mod}\, 14); \quad 12^7 \equiv 12 (\mathrm{mod}\, 14)$$

Indeed, we get the same numbers, but in a different order.

(e) Are these groups isomorphic to a multiplication group of a smaller modulus?

Yes, this group is isomorphic to the multiplicative group mod 7, since 7 is a prime and would produce the cyclic group of order 6.

7. To really tell if two groups are isomorphic, you can write their tables in such an order that they would be identical if you substituted them in place. Why is it helpful to first note the periods and orbits of each element?

This is helpful because then you know which elements could possibly correspond, and which elements definitely don't correspond (i.e. the ones with different periods). Also, once you have chosen one correspondence, all the elements in that orbit are determined.

For the suggested problems, you can write your own program (or use my program) to investigate!