Modulo Arithmetic

Introduction

Modulo arithmetic can be thought of as the arithmetic of remainders where the numbers up to the modulus are the remainders of divisions by the modulus of numbers bigger than or equal to the modulus. When the numbers are less than 0, we use the smallest positive remainder. Let us look at examples:

12/12 = 1 r0 where r0 means remainder of 0
25/12 = 2 r1

-23/12 = -1 r -13 not smallest positive remainder
-23/12 = -2 r1

These expressions are written as follows: 12 mod 12=0 25 mod 12=1 -23 mod 12 = 1
When using modular numbers in arithmetic, we can get results that are counter intuitive to what we get with our conventional number system. However, when we apply modular arithmetic to coding applications, these seemingly anomalies make perfect sense. For most students, this is about as far as we go with modular arithmetic in school. However for those students who want to know more or are interested in generating or breaking codes, modular arithmetic becomes an advanced topic.

Let us look at some examples:
(5+8) mod 12 – 13 mod 12 = 1 for 13/12= 1 r1
5x5 mod 12 = 25 mod 12 = 1 for 25/12=2 r1

From now on, we are doing to leave out the mod operation and use the $\equiv$. Therefore, we have:

$8+5 \equiv 1$
$5x5 \equiv 1$

Prime numbers

Prime numbers play a very important role in modular systems. Prime numbers cannot be factored into other numbers other than themselves and the number 1. Thus prime numbers are cannot give integer results when divided by other numbers. In a modular twelve system, look at what results we get when we keep adding a number to itself.

2: 0, 2, 4, 6, 8, 10, 0
3: 0, 3, 6, 9, 0
4: 0, 4, 8, 0
6: 0, 6, 0
8: 0, 8, 4, 0
9: 0, 9, 6, 3, 0
10: 0, 10, 8, 6, 4, 2, 0

5: 0, 5, 10, 3, 8, 1, 6, 11, 4, 9, 2, 7, 0
7: 0, 7, 2, 9, 4, 11, 6, 1, 8, 3, 10, 5, 0

Only the prime numbers to touch every integer number in the set.

Now look at a modulus system that has a prime number as the modulus.

2: 0, 2, 4, 6, 1, 3, 5, 0
3: 0, 3, 6, 2, 5, 1, 4, 0
4: 0, 4, 1, 5, 2, 6, 3, 0
5: 0, 5, 3, 1, 6, 4, 2, 0
6: 0, 6, 5, 4, 3, 2, 1, 0

Every number touches every number in the set. Also notice the following:

2x4≡8≡1
3x5≡14≡1
6x6≡36≡1

In the base 12 system only:

5x5≡25≡1
7x7≡49≡1

We conclude that if a number has a factor in common with the modulus, that it cannot be multiplied by any other integer to produce a one as the results.

Division

Since division is inverse multiplication, in our modulo 7 as an example, we notice:

1 /2= 4
1/3 = 5
1 /4= 2
1/5 =3
1/6= 6

Thus a fraction is equivalent to an integer. Let us try fraction arithmetic:

5 /6 ≡ **5x6 ≡ 2** where 30 mod 7 =4x7+2 mod 7 = 2
1 /2 + 1/3 = 3/3 x 1/ 2 + 2 / 2 1/3 = 3/6 + 2/6 =**5 / 6**, but
1 /2 + 1/3 ≡ 4+5≡9≡**2**

In multiplication:

1 /2 x 1/3 ≡ 1/6 = 6
1 /2 x 1/3≡4x5≡20≡6

Multiple answers

Look at our modulo 12 system again. 12 modulo 12=0.

1/3 ≡ 1/3, 4 1/3, 8 1/3 because:

3x1/3 ≡ 1 3x 4 1/3 ≡ 13≡1 3x8 1/3 ≡ 25≡1

0/3 = 0, 4, 8 because:

$3x0 \equiv 0 \quad 3x4 \equiv 12 \equiv 0 \quad 3x8 \equiv 24 \equiv 0$

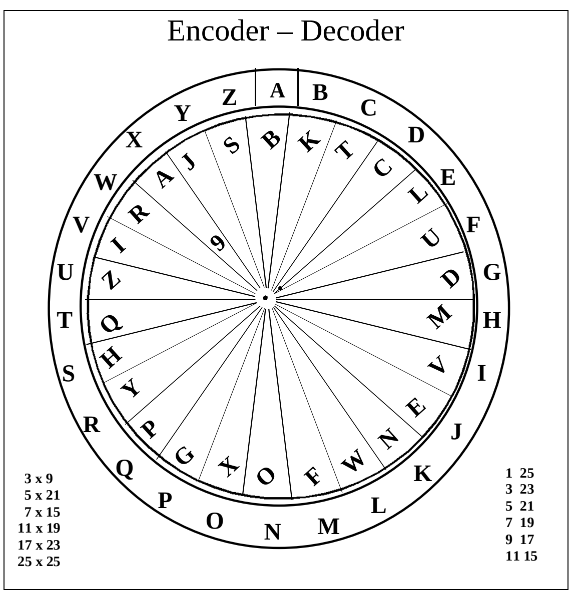
In modular arithmetic, division by an integer gives the number of answers equal to the value of the integer. In modular arithmetic, multiplication by zero does not always give zero as an answer. While modular arithmetic gives us unexpected results, it is consistent with the rules of conventional arithmetic.

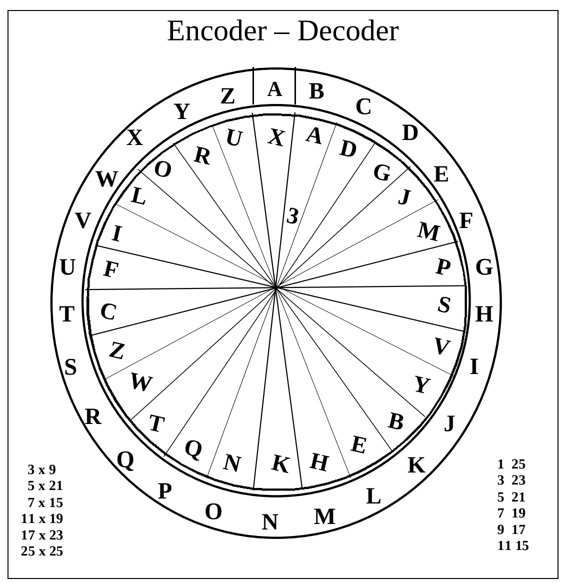
Application

Since we are dealing with modular arithmetic we use a circle rather a number line to represent our application. We assign the following numbers to the letters:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

The encoder on the left is use to code our words and the one on the right is used to decode. We are going to encode the word "HELP". The encode disk is set for $9L+1 \equiv C$.
The decode disk is $L \equiv 3C-3 \equiv 3C+23$. The decode constant was found by dividing the code disk number into 27 using modular arithmetic where 27/9=3. The constant digit is calculated by finding the number of the position under the A on the outer circle.





Let us encode HELP:

H=7 $\quad 9x\,7+1 \equiv 64 \equiv 12$ = M

E=4 $\quad 9x\,4+1 \equiv 37 \equiv 11$ = L

L=11 $\quad 9x11+1 \equiv 100 \equiv 22$ = W

P=15 $\quad 9x15+1 \equiv 136 \equiv 6$ = G

Now let us decode MLWG:

$M=12 \quad 3\times12+23 \equiv 7=H$

$L=11 \quad 3\times11+23 \equiv 4=E$

$W=22 \quad 3\times22+23 \equiv 11=L$

$G=6 \quad 3\times 6+23 \equiv 15=P$

Since we are dealing with a non-prime number, we only have 11 disks for decoding and decoding. This gives us 11x25=275 different codes that we can use. The coding formula is $C\equiv aL+b$ where a=0, 3, 5, 6, 9, 11, 15, 17, 19, 23, and 25 and b is the numbers from 0 to 25. When a=b=0, there is no coding, so we only have 274 codes.

If a person knows two of the letters he can figure out the coding formula. We know that In this case M goes H, and L goes to E. Since $L\equiv aC+b$:

$$7\equiv 12a+b$$

$$4\equiv 11a+b$$

Subtracting, $3\equiv a$

Substituting, $4=11\times3+b$

$$b=4-33=-29=23$$

The decode formula is: $L\equiv 3C+23$

To find the coding formula, we have $9\times L\equiv 9\times3C+9\times23$

$$9L\equiv C+9\times23\equiv C+25$$

$$C=9L-25\equiv 9L+1$$

Manually, the coding disks make it easy to code without having to do the calculations. We could have written out the table, but there are 274 of them. For this example, the table is:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	K	T	C	L	U	D	M	V	E	N	W	F	O	X	G	P	Y	H	Q	Z	I	R	A	J	S

The power of mathematics is very nicely exhibited in this simple problem. This coding scheme is call Affine. When the multiplicative coefficient is 0, it becomes Caesar.