Modular Arithmetic

Let us take a look at some of the **basic rules and properties** that can be applied in Modular Arithmetic(Addition, Subtraction, Multiplication etc.).
Consider numbers **a** and **b** operated under modulo **M**.

1. (a + b) mod M = ((a mod M) + (b mod M)) mod M.
2. (a - b) mod M = ((a mod M) - (b mod M)) mod M.
3. (a * b) mod M = ((a mod M) * (b mod M)) mod M.

The above three expressions are valid and can be performed as stated. But when it comes to modular division, there are some limitations.

There isn't any formula to calculate:

```
(a / b) mod M
```

For this we have to learn **modular inverse**.

Modular Inverse

The modular inverse is an integer 'x' such that.

```
a x ≡ 1 (mod M)
```

The value of x should be in {0, 1, 2, ... M-1}, i.e., in the ring of integer modulo M.

The multiplicative inverse of "a modulo M" exists if and only if a and M are relatively prime (i.e., if gcd(a, M) = 1).

Examples:

```
Input:  a = 3, M = 11
Output: 4
Since (4*3) mod 11 = 1, 4 is modulo inverse of 3
One might think, 15 also as a valid output as "(15*3) mod 11"
is also 1, but 15 is not in ring {0, 1, 2, ... 10}, so not
valid.

Input:  a = 10, M = 17
Output: 12
Since (10*12) mod 17 = 1, 12 is modulo inverse of 3
```

Methods of finding Modular Inverse: There are two very popular methods of finding modular inverse of any number **a** under modulo **M**.

1. **Extended Euclidean Algorithm**: This method can be used when **a** and **M** are co-prime.
2. **Fermat Little Theorem**: This method can be used when **M** is prime.

Let us look at each of the above two methods in details:

Extended Euclidean algorithm that takes two integers 'a' and 'b', finds their gcd and also find 'x' and 'y' such that,

```
ax + by = gcd(a, b)
```

To find modulo inverse of 'a' under 'M', we put b = M in the above formula. Since we know that a and M are relatively prime, we can put value of gcd as 1.

So, the formula becomes:

```
ax + My = 1
```

If we take modulo M on both sides, we get:

```
ax + My ≡ 1 (mod M)
```

We can remove the second term on left side, as 'My (mod M)' would always be 0 for an integer y.

Therefore,

```
ax  ≡ 1 (mod M)
```

So the 'x' that we can find using Extended Euclid Algorithm is modulo inverse of 'a'.

Fermat Little Theorem: The Fermat's little theorem states that if M is a prime number, then for any integer a, the number $\mathbf{a^M - a}$ is an integer multiple of M.

That is,

$$a^M \equiv a \pmod{M}.$$

Since, a and M are co-prime to each other then a^{M-1} is an integral multiple of M.
That is,

$$a^{M-1} \equiv 1 \pmod{M}$$

If we multiply both sides by a^{-1}, we get:

$$a^{-1} \equiv a^{M-2} \bmod M$$

Therefore, if **M is a prime number** to find modulo inverse of **a under M**, find **modular exponentiation of a^{M-2} under modulo M**.