# Modular Arithmetic

### Definition:

* Let , we define to be the remainder when is divided by .

Exercise:



* are **congruent modulo n**, written , if
* Equivalently, IFF

#### Modular Equivalences

* Let and be integers and suppose . The following statements are all equivalent:
* for some integer

Exercise:

True or false?



Exercise:

Find such that .



Exercise:

If , what can we say about ?



## Theorem (Congruence Arithmetic):

* Let , if and , then:

1. ;
2. ;
3. ;
4. ;

### Proof:



Exercise:

1. Given that , find
2. Given that , find
3. Find such that



Exercise:

Find the remainder when is divided by .



### Theorem (Cancellation Law):

* Let
* If and , then .

### Proof:



* NOTE:



Counterexample:



Exercise:

Given , find the smallest such that .



## Congruence Classes modulo n

* The quotient-remainder theorem gives us the following:

#### Fact:

* Let .
* Every integer is congruent modulo to exactly one element in
* This allows us to group integers according to their remainders after dividing by .

### Definition:

* Let .
* The **congruence class (residue)** of is the set .

Exercise:

Write the congruence classes for . How many of them are there?



## Theorem:

* Let . There are exactly distinct congruence classes: .

### Proof:

First, show that no two of are congruent modulo n.

Let .

Then and .

Thus, , so .

Therefore, no two of are congruent, and we have that are all distinct residues.

Next, show that every is in one of these residues.

The Quotient-Remainder Theorem gives .

So, , and .

Therefore, every is in one of .

## Definition:

* Let . The complete set of residues modulo n is the set.

Exercise:

.

In , we have

Exercise:

In ,



## Operations on

* We want to define addition and multiplication on .
* Since different numbers can give the same residue, we must be careful with the definitions.

### Theorem:

* Let .   
  The operation :

is well-defined addition on , i.e. if and , then .

Similarly, the operation :

is well-defined multiplication on , i.e. if and , then .

### Proof:

Exercise:

Write addition and multiplication tables for .



|  |  |  |  |
| --- | --- | --- | --- |
|  | [0] | [1] | [2] |
| [0] |  |  |  |
| [1] |  |  |  |
| [2] |  |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
|  | [0] | [1] | [2] |
| [0] |  |  |  |
| [1] |  |  |  |
| [2] |  |  |  |

### Properties of

1. and are closed (binary) operations.
2. and are commutative
3. and are associative
4. is distributive over
5. Identities are under , under
6. The additive inverse of is
7. Multiplicative inverses exist only for

### Theorem:

* If and are coprime, then there exists such that .
* We call the multiplicative inverse of .
* is unique modulo .
* We write .

#### Proof:

Consider the set .

If we can show that there are all distinct modulo , then exactly one of them is equal to .

Suppose the contrary: .

Then , so .

But and are coprime, so .

This is a contradiction, since and are distinct nonnegative integers less than .

Exercise:

Find a multiplicative inverse of .



Exercise:

Find .



## Application: Cryptography

* Cryptography is the study of methods for sending secrete messages.
* There are many techniques for encryption and decryption, one of which is **public-key cryptography**.
* The method uses big prime numbers and modular arithmetic.
* **RSA** is one such public-key method.

### RSA

1. Choose 2 large primes .
2. Choose that is coprime with .
3. Choose .
4. The public key is . This is available to everyone for encryption.
5. The private key is . This is available only to those who the send wants to be able to decrypt.

#### Encryption Step:

* Let the message to be encrypted be (a computer uses binary code for everything, so encrypting integers is sufficient).
* The encrypted message is .

#### Decryption Step:

* is received by .
* We will not see the proof.
* and are chosen to be several hundred digits long each, making it impossible for a computer to finds the factors in reasonable time.
* We will see some examples with small primes.

Example:

Let , public key (3, 55). Encrypt and decrypt the message “HEY.”



Exercise:

Decrypt the message 41 83 36 that was encrypted with public key .

