

Homework Number: 03
 Name: Yi Qiao
 ECN Login: qiao22
 Due Date: 1/31/2019

Theory Problems

1.

If the two operator is switched, it will no longer be a ring for the following reasons:

1. Assuming the set is \mathbb{R} , multiplication does not have a identity element in the set.

2. Even if we choose a better set, for example $(0, \infty]$, which have a identity element for multiplication, the ring operator, $(+)$ in this case does not distribute over the group operator (\times) .

2.

Euclid's Algorithm:

$$\begin{aligned}
 \gcd(1344, 752) &= \gcd(752, 1344 \% 752) &= \gcd(752, 592) \\
 &= \gcd(592, 752 \% 592) &= \gcd(592, 160) \\
 &= \gcd(160, 592 \% 160) &= \gcd(160, 112) \\
 &= \gcd(112, 160 \% 112) &= \gcd(112, 48) \\
 &= \gcd(48, 112 \% 48) &= \gcd(48, 16) \\
 &= \gcd(16, 48 \% 16) &= \gcd(16, 0) \\
 &= 16
 \end{aligned} \tag{1}$$

Stein's Algorithm:

$$\begin{aligned}
 \gcd(1344, 752) &= 2 * \gcd(672, 376) \\
 &= 4 * \gcd(336, 188) \\
 &= 8 * \gcd(168, 94) \\
 &= 16 * \gcd(84, 47) \\
 &= 16 * \gcd(42, 47) \\
 &= 16 * \gcd(21, 47) \\
 &= 16 * \gcd(13, 21) \\
 &= 16 * \gcd(4, 13) \\
 &= 16 * \gcd(2, 13) \\
 &= 16 * \gcd(1, 13) \\
 &= 16 * \gcd(6, 1) \\
 &= 16 * \gcd(3, 1) \\
 &= 16 * \gcd(1, 1) \\
 &= 16
 \end{aligned} \tag{2}$$

3.

Suppose there exist $\alpha = 25^{-1}$ and a ring identity 1 in Z_{30}

$$\begin{aligned} (\alpha \times 25) \bmod 30 &= 1 \\ \alpha \times 25 &= 30 \times n + 1 \text{ where } n \in \mathbb{Z} \\ \alpha &= \frac{6}{5} \times n + \frac{1}{25} \end{aligned} \tag{3}$$

By inspection, there does not exist such a $n \in \mathbb{Z}$ making $\alpha \in \mathbb{Z}$

Thus, 25 does not have a multiplicative inverse in Z_{30}

4.

$$\begin{aligned} \gcd(33, 23) &= \gcd(23, 10) \quad \text{residue } 10 = 1 \times 33 - 1 \times 23 \\ &= \gcd(10, 3) \quad \text{residue } 3 = 1 \times 23 - 2 \times 10 \\ &= 1 \times 23 - 2 \times (1 \times 33 - 1 \times 23) \\ &= 3 \times 23 - 2 \times 33 \\ &= \gcd(3, 1) \quad \text{residue } 1 = 1 \times 10 - 3 \times 3 \\ &= 1 \times (1 \times 33 - 1 \times 23) - 3 \times (3 \times 23 - 2 \times 33) \\ &= 7 \times 33 - 10 \times 23 \end{aligned} \tag{4}$$

Therefore, the inverse of 23 in Z_{33} is 23

5.

(a)

$$8x \equiv 5 \pmod{23}$$

Since 8 is prime relative to 23, there exist a multiplicative inverse in Z_{23} for 8, which can be obtained by the Extended Euclid's algorithm:

$$\begin{aligned} \gcd(23, 8) &= \gcd(8, 7) \quad \text{residue } 7 = 1 \times 23 - 2 \times 8 \\ &= \gcd(7, 1) \quad \text{residue } 1 = 1 \times 8 - 1 \times 7 \\ &= 1 \times 8 - 1 \times (1 \times 23 - 2 \times 8) \\ &= 3 \times 8 - 1 \times 23 \end{aligned} \tag{5}$$

Thus, 8^{-1} in Z_{23} is 3

$$\begin{aligned} 8x &\equiv 5 \pmod{23} \\ x &= 5 \times 8^{-1} = (5 \times 3) \bmod 23 \\ &= 15 \end{aligned} \tag{6}$$

(b)

$$6x \equiv 3(mod\ 19)$$

For the same rational in (a)

$$gcd(19, 6) = gcd(6, 1) \quad residue\ 1 = 1 \times 19 - 3 \times 6 \quad (7)$$

Thus, 6^{-1} in Z_{19} is $-3\ mod\ 19 = 16$

$$\begin{aligned} 6x &\equiv 3(mod\ 19) \\ x &= 3 \times 6^{-1} = 3 \times 16\ mod\ 19 \\ &= 10 \end{aligned} \quad (8)$$

(c)

$$25x \equiv 9(mod\ 7)$$

For the same rational in (a)

$$\begin{aligned} gcd(25, 7) &= gcd(7, 4) \quad residue\ 4 = 1 \times 25 - 3 \times 7 \\ &= gcd(4, 3) \quad residue\ 3 = 1 \times 7 - 1 \times 4 \\ &= 1 \times 7 - 1 \times (1 \times 25 - 3 \times 7) \\ &= 4 \times 7 - 1 \times 25 \\ &= gcd(3, 1) \quad residue\ 1 = 1 \times 4 - 1 \times 3 \\ &= (1 \times 25 - 3 \times 7) - (4 \times 7 - 1 \times 25) \\ &= 2 \times 25 - 7 \times 7 \end{aligned} \quad (9)$$

Thus, $25^{-1} = 4^{-1}$ in Z_7 is 2

$$\begin{aligned} 25x &\equiv 9(mod\ 7) = 2 \\ x &= 2 \times 4^{-1} = 2 \times 2\ mod\ 7 \\ &= 4 \end{aligned} \quad (10)$$

Programming Question

Since if Z_n is a field, n has to be prime, otherwise, we can always find some integer $n \in [0..n-1]$ who does not have a multiplicative inverse in Z_n . So, the program is essentially judging the input is prime or not.

Code

```
#!/usr/bin/env python3

def _isPrime(n):

    for i in range(2, int(n**(1/2))):
        if n % i == 0:
            return False
    return True

if __name__ == "__main__":

    n = int(input("Please Enter a number:"))
    if _isPrime(n):
        print("field")
    else:
        print("ring")
```