

## บทที่ 1 ทฤษฎีจำนวน (Number Theory) สัปดาห์ที่ 2

ในสัปดาห์นี้เรามาทำความรู้จักกับการเขียนโปรแกรมเพื่อป้อนข้อมูลผ่านแป้นพิมพ์ การเปลี่ยนชนิดข้อมูล การตรวจสอบ modular congruence การคำนวณหาค่า ห.ร.ม.(gcd) ค.ร.น. (lcm) และการประยุกต์ใช้ทฤษฎีจำนวน เช่น การสร้างเลขสุ่มเทียม

### 1. การป้อนข้อมูลผ่านแป้นพิมพ์ (Input data from keyboard)

ผู้ใช้สามารถใส่ค่าที่ต้องการได้ผ่านทางแป้นพิมพ์โดยทำงานผ่านฟังก์ชันที่ชื่อ input การรับข้อมูลผ่านทางแป้นพิมพ์ด้วยฟังก์ชัน input ในภาษา Python จะรับข้อมูลเป็นชนิด String เสมอ มีรูปแบบการใช้งานดังนี้

```
var = input([prompt])
```

- var ตัวแปร
- input เรียกใช้ฟังก์ชัน input
- prompt ข้อความแจ้งให้ผู้ใช้ใส่ข้อมูล

สังเกตว่า prompt อยู่ใน [ ] หมายถึงผู้ใช้จะใส่ข้อความหรือไม่ใส่ก็ได้

**ตัวอย่าง 1** แสดงการรับข้อมูลจากแป้นพิมพ์

```
1: varI = input('Input Data from keyboard (Integer):')
2: print(varI)
3:
4: varS = input('Input Data from keyboard (String):')
5: print(varS)
```

**ผลลัพธ์**

```
Input Data from keyboard (Integer): 24
24
Input Data from keyboard (String): Buu
Buu
```

**อธิบาย**

บรรทัด 1 สร้างตัวแปร varI ไว้เก็บข้อมูลที่ผู้ใช้ป้อนทางแป้นพิมพ์ ในที่นี้ระบุให้ใส่ข้อมูลชนิด Integer

บรรทัด 2 แสดงผลข้อมูลที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์

บรรทัด 4 สร้างตัวแปร varS ไว้เก็บข้อมูลที่ผู้ใช้ป้อนทางแป้นพิมพ์ ในที่นี้ระบุให้ใส่ข้อมูลชนิด String

บรรทัด 5 แสดงผลข้อมูลที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์

หลักการทำงาน คือ บรรทัด 1 ถูกเรียกทำงาน ซึ่งจะแสดงข้อความ *Input Data form keyboard (Integer):* และให้ผู้ใช้ใส่ข้อมูล อาจจะเป็นตัวเลข หรือ ข้อความ เช่น 24 รับเข้ามานั้น จะถูกเก็บไว้ในตัวแปร varI ที่กำหนด บรรทัดต่อมาใช้คำสั่ง print เพื่อแสดงค่าที่อยู่ในตัวแปร varI ออกทางหน้าจอ

## 2. การเปลี่ยนชนิดข้อมูล

เนื่องจากฟังก์ชัน input รับข้อมูลผ่านทางแป้นพิมพ์เป็นชนิด String การที่จะนำข้อมูลชนิด String ไปใช้คำนวณทางคณิตศาสตร์จำเป็นจะต้องแปลงค่าให้ข้อมูลที่รับทางแป้นพิมพ์ที่เป็นชนิด String นั้น ให้เป็น ชนิด Integer ก่อน โดยมีรูปแบบการแปลงข้อมูลดังนี้

**ตัวอย่าง 2** การแปลงข้อมูลจาก String เป็น Integer

```
1: a = input('Input number: ')
2: sum = int(a) + 13
3: print(sum)
```

**ผลลัพธ์**

```
Input number: 5
18
```

**อธิบาย**

บรรทัด 1 สร้างตัวแปร a ใช้เก็บข้อมูลที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์  
บรรทัด 2 จากข้อมูลที่ได้รับมาในตัวแปร a นั้น เป็นข้อมูลชนิด String จะนำมาคำนวณทางคณิตศาสตร์จะต้องนำมาแปลงเป็น Integer ผ่านฟังก์ชัน int() เก็บไว้ในตัวแปร sum  
บรรทัด 3 แสดงข้อมูลที่อยู่ในตัวแปร sum ผ่านทางหน้าจอ  
หลักการทำงาน คือ ผู้ใช้ป้อนข้อมูลทางแป้นพิมพ์เป็นตัวเลข มาเก็บไว้ในตัวแปร a และทำการเปลี่ยนชนิดข้อมูลจาก String เป็น Integer และใช้คำสั่ง print เพื่อแสดงค่าที่คำนวณเสร็จแล้วผ่านทางหน้าจอ

**ตัวอย่าง 3**

```
1: a = int(input('Input number: '))
2: sum = a + 13
3: print(sum)
```

**ผลลัพธ์**

```
Input number: 5
18
```

**อธิบาย**

บรรทัด 1 สร้างตัวแปร a ใช้เก็บข้อมูลที่ผู้ใช้ป้อนผ่านทางแป้นพิมพ์ โดยมีการเปลี่ยนชนิดของข้อมูลก่อนทำการตั้งค่าให้กับตัวแปร a  
บรรทัด 2 เนื่องจาก a ได้ถูกเปลี่ยนค่าเป็นตัวเลขแล้วใน บรรทัดที่ 1 ทำให้สามารถนำ a มาคำนวณทางคณิตศาสตร์ได้เลย  
บรรทัด 3 คำสั่ง print(sum) แสดงข้อมูลที่อยู่ในตัวแปร sum ผ่านทางหน้าจอ

## 3. การหาตัวหารร่วมมาก(ห.ร.ม. หรือ gcd)

**ตัวอย่าง 4** จงหา gcd(24,36)

```
1: #Find gcd(a,b)
2: a = int(input("Enter a : "))
3: b = int(input("Enter b : "))
4: while b>0:
5:     print('a = ', a, ' b = ', b, ' r = ', a%b)
6:     a, b = b, a % b
7: print('gcd(a,b) = ',a)
```

### ผลลัพธ์

```
Enter a : 24
Enter b : 36
a = 24 b = 36 r = 24
a = 36 b = 24 r = 12
a = 24 b = 12 r = 0
gcd(a,b) = 12
```

### ตัวอย่าง 5 จงหา gcd(35,60)

#### ผลลัพธ์

```
Enter a : 35
Enter b : 60
a = 35 b = 60 r = 35
a = 60 b = 35 r = 25
a = 35 b = 25 r = 10
a = 25 b = 10 r = 5
a = 10 b = 5 r = 0
gcd(a,b) = 5
```

### ตัวอย่าง 6 จงหา gcd(35,12)

#### ผลลัพธ์

```
Enter a : 35
Enter b : 12
a = 35 b = 12 r = 11
a = 12 b = 11 r = 1
a = 11 b = 1 r = 0
gcd(a,b) = 1
```

## 4. การหาตัวคูณร่วมน้อย(ค.ร.น. หรือ lcm)

### ตัวอย่าง 7 จงหา lcm(24,36)

```
#'Find gcd(a,b) and lcm(a,b)
a = int(input("Enter a : "))
b = int(input("Enter b : "))
x = a
y = b
while b>0:
    print('a = ', a, ' b = ', b, ' r = ', a%b)
    a, b = b, a % b
print('gcd(a,b) = ', a)
print('lcm(a,b) = ', (x*y)//a)
```

#### ผลลัพธ์

```
Enter a : 24
Enter b : 36
a = 24 b = 36 r = 24
a = 36 b = 24 r = 12
a = 24 b = 12 r = 0
gcd(a,b) = 12
lcm(a,b) = 72
```

## 5. การตรวจสอบ Modular Congruence

เมื่อกำหนดให้  $a, b \in \mathbb{Z}$ ,  $m \in \mathbb{Z}^+$

$a$  คอนกรูเอนซ์(congruent) กับ  $b$  มอดุโล  $m$ , เขียนได้ว่า " $a \equiv b \pmod{m}$ ", ก็ต่อเมื่อ  $m \mid a-b$   
เรียก  $m$  ว่า มอดุลัส หรือเขียนได้ว่า:  $(a-b) \bmod m = 0$

ตัวอย่าง 8 การตรวจสอบ  $172 \equiv 177 \pmod{5}$

```
1: print('Check that "A congruence B (mod m)"')
2: A = int(input("Enter a number of A: "))
3: B = int(input("Enter a number of B: "))
4: m = int(input("Enter a number of m: "))
5: if (m > 0):
6:     if ((A-B)%m) == 0:
7:         print(A, ' congruence ', B, '(mod ', m, ')')
8:     else:
9:         print(A, ' is not congruence ', B, '(mod ', m, ')')
```

ผลลัพธ์

```
Check that "A congruence B (mod m)"
Enter a number of A: 172
Enter a number of B: 177
Enter a number of m: 5
172 congruence 177 (mod 5 )
```

ตัวอย่าง 9 การตรวจสอบ  $9 \equiv -18 \pmod{11}$

ผลลัพธ์

```
Check that "A congruence B (mod m)"
Enter a number of A: 9
Enter a number of B: -18
Enter a number of m: 11
9 is not congruence -18 (mod 11 )
```

## 6. การสร้างเลขสุ่มเทียม

การสร้างตัวเลขสุ่มแบบเทียมโดยใช้คอนกรูเอนซ์ เริ่มด้วยการเลือกจำนวนเต็มบวก 4 จำนวน ได้แก่ :  
มอดุลัส(modulus)  $m$ , พหุคูณ(multiplier)  $a$ , ค่าที่เพิ่มขึ้น(increment)  $c$ , และ ค่าเริ่มต้น(seed)  $x_0$  โดยที่  
 $2 \leq a < m$ ,  $0 \leq c < m$ ,  $0 \leq x_0 < m$  เพื่อทำการสร้างลำดับเลขสุ่มเทียม  $\{x_n\}$  ซึ่ง  $0 \leq x_n < m$  โดยใช้ :

$$x_{n+1} = (ax_n + c) \bmod m$$

ตัวอย่าง 10 การสร้างเลขสุ่มเทียม 4 จำนวน

```
1: seed = 426
2: a = 293 #multiple
3: c = 467 #increment
4: m = 1000 #modulus
5: num = int(input("Enter a number of random number : "))
6: rn = seed
7: for i in range(1, num+1):
8:     rn = ((a * rn) + c) % m
9:     print(rn)
```

### ผลลัพธ์

Enter a number of random number : 4

285

972

263

526