



Mahidol University  
*Wisdom of the Land*



# Lecture 06: Repetition - II

Lecturers:

Aj. Jidapa Kraisangka

Aj. Akara Supratak

Aj. Tipajin Thaipisutikul



- **RECAP:** `while` and `for` statement
- **More about Repetition:**
  - Loop control statements: `continue` and `break`
  - Loop in Loop (Nested Loop)



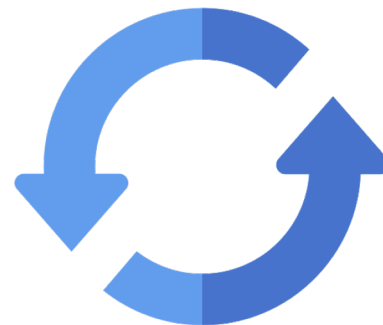
- **3 Session** ในช่วงก่อนกลางภาคการศึกษา ดังนี้
    - **Friday, Sep 16 4:00-6:00pm**
    - **Wed, Sep 21: 4:00-6:00pm**
    - **Friday, Sep 23: 4:00-6:00pm**
- @ Lab106**



- **Flow controls (in programming)** เป็นลักษณะการเขียนโปรแกรม ที่มีลำดับการประมวลผลแตกต่างกันไป
- รูปแบบของ Flow controls ได้แก่
  - **Sequential:** รูปแบบปกติที่เป็นลำดับขั้นตอนเรียงกัน
  - **Selection:** รูปแบบที่การเลือกทำชุดคำสั่งบางชุดตามเงื่อนไข
  - **Repetition:** รูปแบบปกติที่ทำซ้ำในบางชุดคำสั่ง
  - **Invocation:** รูปแบบการเรียกใช้ชุดคำสั่งอื่น ๆ หรือฟังก์ชัน



- **Repetition** หรือการเขียนโปรแกรมแบบวนซ้ำ
- เป็นการนำชุดคำสั่งมาทำงานซ้ำหลายๆ รอบ
- จะทำงานที่รอบขึ้นอยู่กับการเงื่อนไขที่ผู้เขียนโปรแกรมกำหนดไว้
- ชุดคำสั่งที่มีการวนใช้ซ้ำๆ เรียกว่า **loop**





- ควบคุมโปรแกรมให้ทำงานบางอย่างซ้ำๆ ในขณะที่เงื่อนไข (expression) ของ loop นั้นยังคงเป็นจริง (ไม่เท่ากับ 0) อยู่
- มักถูกใช้เมื่อจำนวนรอบในการวน loop ไม่ทราบล่วงหน้า

```
while expression:  
    statement(s)
```

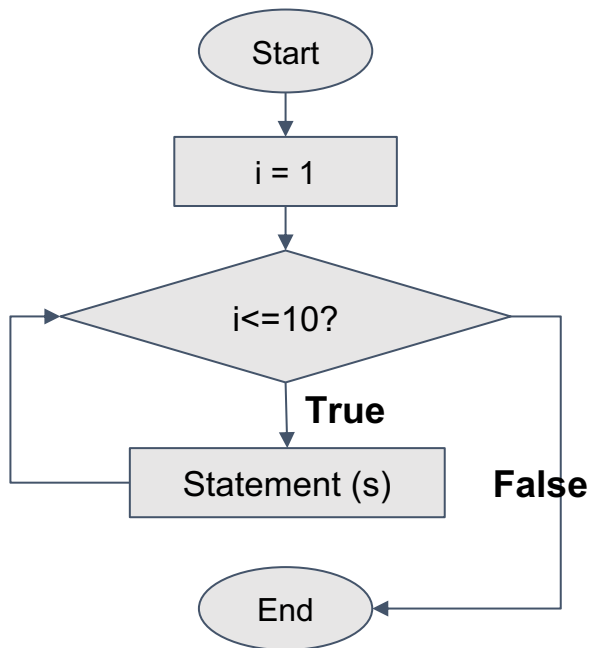
ขั้นตอนการทำงานของ while loop:

1. Test the expression
2. If the expression is TRUE
  - a. Execute the statement(s)
  - b. Go to step 1

Else exit the while loop



Print เลข 1 ถึง 10



Initialization

Conditions จะต้องเป็น **True**  
เพื่อให้ loop ทำงานต่อไป

```
i = 1
while i <= 10:
    print(i)
    i = i + 1
```

Alteration

Output:

1  
2  
3  
4  
5  
...  
10



- เป็นคำสั่งวนซ้ำที่ใช้ควบคุมการทำงานซ้ำๆ มักใช้สำหรับการวนอ่านค่าใน **iterable object** เช่น list, string, tuple
- มักใช้เมื่อ**ทราบจำนวนรอบ**ในการวน loop แน่แน่นอน

```
for var in <sequence>:  
    statement(s)
```

## ขั้นตอนการทำงาน for loop:

1. Item แรกใน sequence ถูก assign เก็บใน **var**
  2. ชุดคำสั่งใน for loop ถูก execute
  3. If item ใน sequence ยังมีอยู่
    - a. item ถัดมาใน sequence จะถูก assign ให้ **var**
    - b. กลับไป step 2
- Else ออกจาก for loop





- **range ( )** เป็น built-in function ที่มักใช้ร่วมกับ for loop
- ใช้ในการสร้าง sequence ของตัวเลข ภายใน range ที่กำหนด
- Default: เริ่มต้นที่ 0, เพิ่มขึ้นทีละ 1, และจบที่ตัวเลขสุดท้ายที่กำหนด

## Syntax:

**range(stop)**

**range(start, stop)**

**range(start, stop, step)**

มี parameter 3 ตัว เป็น type integer ได้แก่

- **start:** ตัวเลขเริ่มต้น
- **stop:** ตัวเลขสุดท้าย (ไม่รวม)
- **step:** ค่าที่เปลี่ยนแปลง

ไม่ support float numbers



- **for:** มักใช้เมื่อทราบจำนวนรอบในการ loop อย่างแน่นอน
  - ใช้ร่วมกับ function `range()` และ iterable objects เช่น list, string, tuple
- **while:** มักใช้เมื่อจำนวนรอบในการ loop ไม่ทราบล่วงหน้า
  - เช่น โปรแกรมที่จำนวน loop ขึ้นกับ user input
  - แต่สามารถใช้กับกรณีทราบจำนวนรอบในการ loop ล่วงหน้าแล้วเช่นกัน ขึ้นอยู่กับ style การ code ของ programmer



Mahidol University  
*Wisdom of the Land*



# Loop Control Statement



- ใช้ภายใน **while** และ **for** loop
- **break:** ใช้สำหรับหยุดการทำงานของ loop ทันทีโดยไม่จำเป็นต้องให้เงื่อนไขเป็น **False** ก่อน
- **continue:** ใช้สำหรับข้ามการทำงานของ loop ไปทำงานในรอบต่อไปโดยไม่สนใจคำสั่งที่เหลือหลังจากนั้น



```
while expression:
    if expression:
        break
    statement1
    statement2
    ...
    statement n
```

หยุดการทำงานของ loop ทันที

```
while expression:
    if expression:
        continue
    statement1
    statement2
    ...
    statement n
```

ข้าม statement ถัด ๆ มาใน loop และ  
ไปทำงานในรอบใหม่ทันที



เขียนโปรแกรมเพื่อรับ input ตัวเลขมากที่สุด 10 จำนวน โดยถ้าตัวเลขที่รับมาเป็นจำนวนลบ ให้ออกจาก loop ทันทีและแสดงผลรวมของ input ที่เป็นจำนวนบวกทั้งหมด

**Output:**

Enter a number :10  
Enter a number :5  
Enter a number :3  
Enter a number :0  
Enter a number :-1  
18

```
total = 0
for x in range(10):
    num = (int)(input("Enter a number :"))
    if num<0:
        break
    total = total + num
print(total)
```



Revisit: เขียนโปรแกรมเพื่อรับ input ตัวเลข จนกว่าตัวเลขที่รับจะเป็นจำนวนลบ จากนั้นแสดงผลรวมของ input ที่เป็นจำนวนบวกทั้งหมด

**Output:**

```
Enter a number :10
Enter a number :5
Enter a number :3
Enter a number :0
Enter a number :-1
18
```

```
total = 0
while True:
    num = (int)(input("Enter a number :"))
    if num >= 0:
        total += num
    else:
        break
print(total)
```

loop ไปเรื่อยๆ

ออกจาก loop เมื่อตัวเลขติดลบ



Print ตัวเลข 1 ถึง 10 ยกเว้น 5

1 2 3 4 6 7 8 9 10

## while Statement

```
x = 0
while x < 10:
    x += 1
    if x == 5:
        continue
    print(x, end=" ")
```

## for Statement

```
for x in range(1, 11):
    if x == 5:
        continue
    print(x, end=" ")
```





# Loop in Loops

(Nested Loop)



Print 1-5 จำนวน 10 รอบ

ต้องใช้ 10 for loop ต่อกัน

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```

```
for x in range(5):
    print(x+1, end=" ")
print()

for x in range(5):
    print(x+1, end=" ")
print()

...

for x in range(5):
    print(x+1, end=" ")
print()
```

หากใช้ Nested Loop จะทำให้ code  
สั้นและกระชับขึ้น

```
for x in range(10):
    for y in range(5):
        print(y+1, end=" ")
    print()
```

Print 10 แถว

แต่ละแถว print 1-5



## Loop สามารถใส่ซ้อนกันได้ (loop ซ้อน loop)

**for/while:**

**for/while:**

statement(s)

**for/while:**

statement(s)

```
for x in range(...):  
    for y in range(...):  
        statement(s)
```

```
while expression1:  
    while expression2:  
        statement(s)
```

```
while expression1:  
    for x in range(...):  
        statement(s)
```



```
for x in range(4): # outer loop
```

```
    print("row", x, end=": ")
```

```
    for y in range(3): # inner loop
```

```
        print("col", y, end=" ")
```

```
    print()
```

Loop นอก: รวน loop ทั้งหมด 4 รอบ

Loop ใน: รวน loop ทั้งหมด 3 รอบ

```
row 0: col 0 col 1 col 2
row 1: col 0 col 1 col 2
row 2: col 0 col 1 col 2
row 3: col 0 col 1 col 2
```



```
i=0
while i<4: # outer loop
    print("row ", i, end=": ")
    j=0
    while j<3: # inner loop
        print("col ", j, end=" ")
        j+=1
    print()
    i+=1
```

แยกตัวแปรเงื่อนไขในการวน loop ภายนอก  
และภายใน (นั่นคือ i and j)

```
row 0: col 0 col 1 col 2
row 1: col 0 col 1 col 2
row 2: col 0 col 1 col 2
row 3: col 0 col 1 col 2
```



สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=4$

```
* * * *  
* * * *  
* * * *  
* * * *
```





สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops = 4 (หรือ  $n$ )    # Inner loops = 4 (หรือ  $n$ )

ตัวอย่าง  $n=4$

```
* * * *  
* * * *  
* * * *  
* * * *
```

$n$  ควบคุมจำนวนแถว  
และจำนวนหลัก

```
n = int(input())  
for i in range(n):  
    for j in range(n):  
        print('*', end=' ')  
    print()
```



สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output  
ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

○	x	○	x	○
○	x	○	x	○
○	x	○	x	○
○	x	○	x	○
○	x	○	x	○





สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

o	o	o	o	o
x	x	x	x	x
o	o	o	o	o
x	x	x	x	x
o	o	o	o	o



สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

O	X	O	X	O
X	O	X	O	X
O	X	O	X	O
X	O	X	O	X
O	X	O	X	O



สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4





สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =  $n$

# Inner loops =  $n$

ตัวอย่าง  $n=5$

0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4
0	1	2	3	4

$n$  ควบคุมจำนวนแถว  
และจำนวนหลัก

```
n = int(input())  
for i in range(n):  
    for j in range(n):  
        print(j, end=" ")  
    print()
```



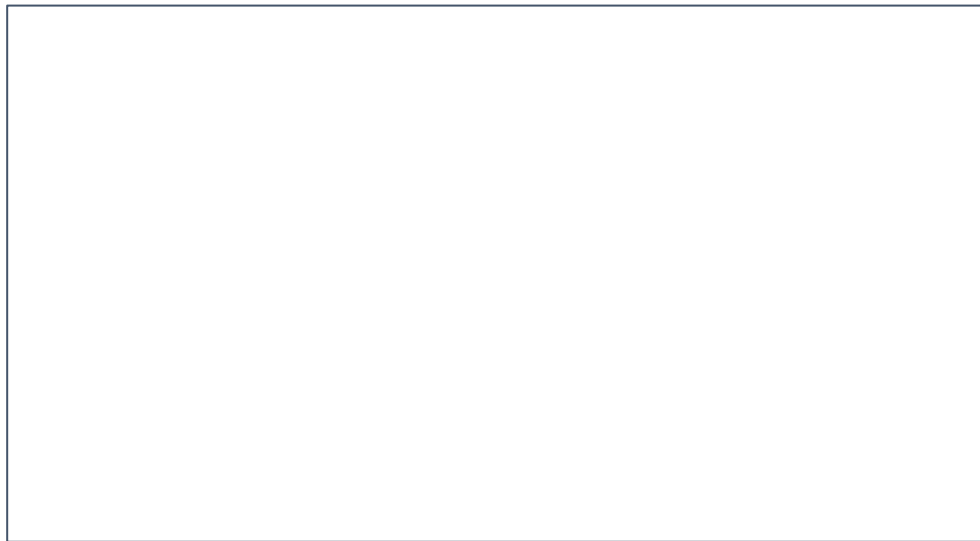
สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

0	0	0	0	0
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4





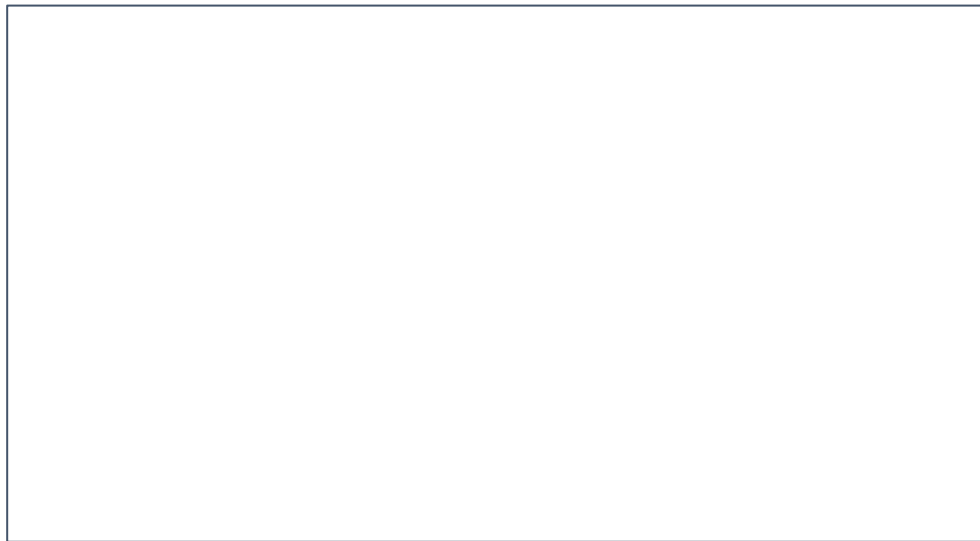
สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=5$

5	4	3	2	1
5	4	3	2	1
5	4	3	2	1
5	4	3	2	1
5	4	3	2	1





สมมติ  $n$  คือจำนวนแถวที่ผู้ใช้ input เขียนโปรแกรมเพื่อ print output ดังต่อไปนี้

# Outer loops =

# Inner loops =

ตัวอย่าง  $n=3$

10	9	8	7	6	5	4	3	2	1
10	9	8	7	6	5	4	3	2	1
10	9	8	7	6	5	4	3	2	1



- ตัวอย่างที่ผ่านมา จำนวนรอบของ inner loop ไม่มีการเปลี่ยนแปลง
- ในแต่ละรอบของ outer loop จำนวนรอบของ inner loop สามารถเปลี่ยนแปลงได้

```
for x in range(n):  
    for y in range(x+1):  
        print(y+1, end=" ")  
    print()
```





## Nested Loop: A More Advanced Example

```
for x in range(n):  
    for y in range(x+1):  
        print(y+1, end=' ')  
        print()
```

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5  
1 2 3 4 5 6  
1 2 3 4 5 6 7
```

สมมติ #outer loop มีค่า  $n = 7$

```
x = 0: y in [0, 1)  #inner loops=1  
x = 1: y in [0, 2)  #inner loops=2  
x = 2: y in [0, 3)  #inner loops=3  
x = 3: y in [0, 4)  #inner loops=4  
x = 4: y in [0, 5)  #inner loops=5  
x = 5: y in [0, 6)  #inner loops=6  
x = 6: y in [0, 7)  #inner loops=7
```



## Nested Loop: A More Advanced Example



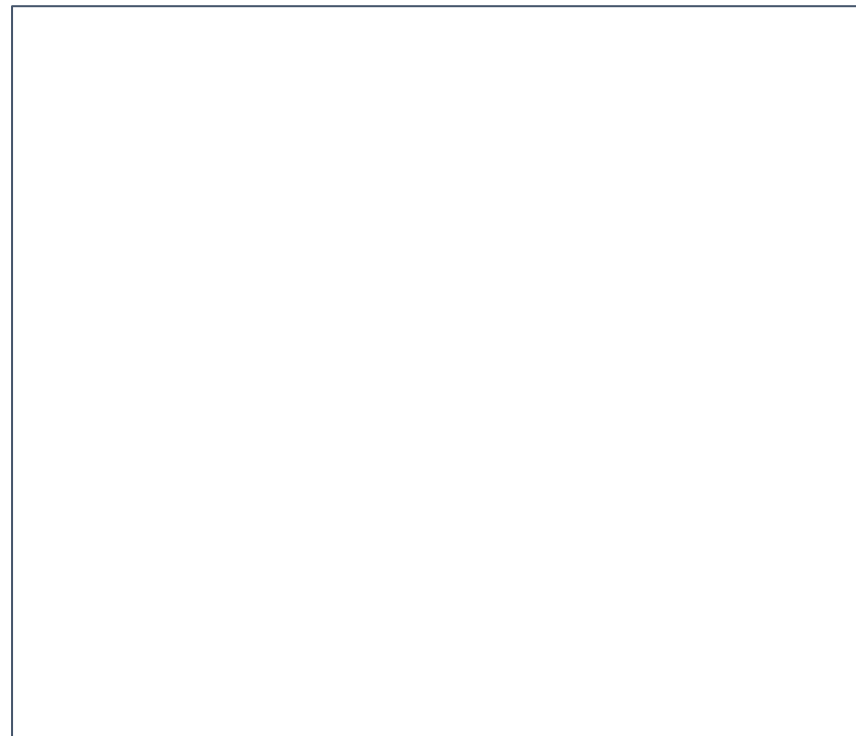
```
for i in range(7, 0, -1):
    for j in range(i):
        print(' * ', end=' ')
    print()
```

[illegible]



Output ของโปรแกรมนี้คืออะไร

```
for i in range(10):  
    if i%2==0:  
        x = '*'  
    else:  
        x = '^'  
    for j in range(i):  
        print(x, end=' ')  
    print()
```





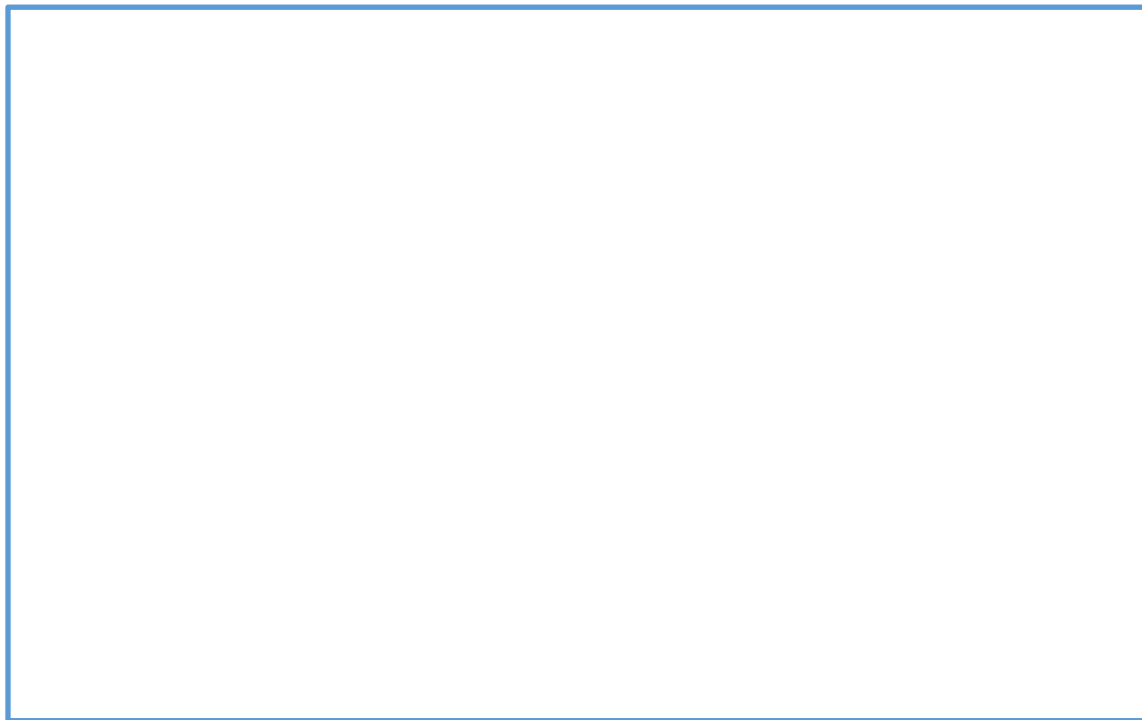
Output ของโปรแกรมนี้คืออะไร

```
for i in range (5):  
    print('o '*i, end='')  
    for j in range(i+1):  
        print('x', end=' ')  
    print()
```



จงเขียนโปรแกรมโดยใช้ เพื่อแสดงตารางสูตรคูณตั้งแต่  
แม่ 2 จนถึงจำนวน  $x$  และจำนวนแถวของตารางสูตรคูณ  $y$   
เช่น  $x=5$ ,  $y=3$

$2 \times 1 = 2$   
 $2 \times 2 = 4$   
 $2 \times 3 = 6$   
 $3 \times 1 = 3$   
 $3 \times 2 = 6$   
 $3 \times 3 = 9$   
 $4 \times 1 = 4$   
 $4 \times 2 = 8$   
 $4 \times 3 = 12$   
 $5 \times 1 = 5$   
 $5 \times 2 = 10$   
 $5 \times 3 = 15$



$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$3 \times 1 = 3$$

$$3 \times 2 = 6$$

$$3 \times 3 = 9$$

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$



**Mahidol University**  
*Wisdom of the Land*



KEEP  
CALM  
AND  
CODE  
ON

[keep-calm.net](https://keep-calm.net)

Thanks to: <https://keep-calm.net/keep-calm-and-code-on.html>

# Happy Coding :)