

w13-Lec

Iterations

Part II

for 204111

Areerat Trongratsameethong
& Kittipitch Kuptavanich

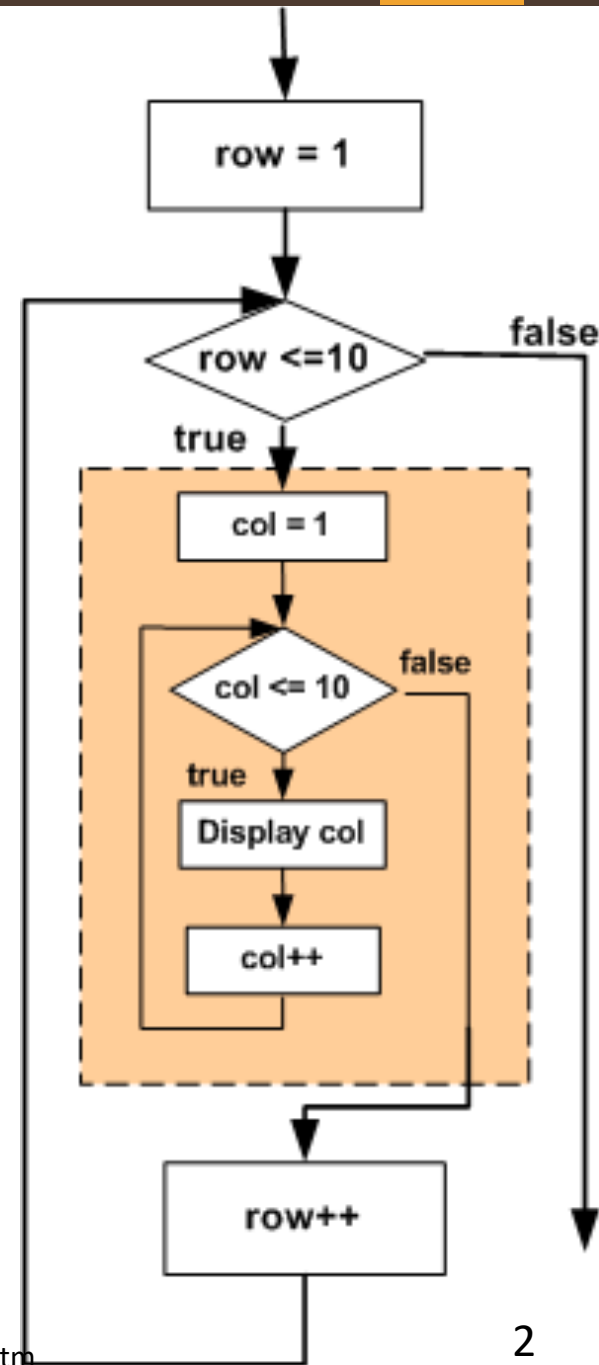
Nested Loops

Pseudocode

Initial Condition: row = 1

```

Outer Loop {
  WHILE row <= 10
    Initial Condition: col = 1
    Inner Loop {
      WHILE col <= 10
        Display col
        Add 1 to col
      }
      Add 1 to row
    }
  ENDWHILE
}
ENDWHILE
  
```



Nested Loops [2]

```

05 def nested_loop():
06     ROW = 5
07     COLUMN = 4
08     for i in range(1, ROW + 1):           # Loop ၁၀၀၈ starts
09         print("\ni is now %d" % i)
10
11     { for j in range(1, COLUMN + 1): # Loop ၁၀၀၈ starts
12         print("j = %d" % j, end=" ")
13     }
...

```

```

i is now 1
j = 1 j = 2 j = 3 j = 4
i is now 2
j = 1 j = 2 j = 3 j = 4
i is now 3
j = 1 j = 2 j = 3 j = 4
i is now 4
j = 1 j = 2 j = 3 j = 4
i is now 5
j = 1 j = 2 j = 3 j = 4

```

Nested Loops [3]

```

i = 1
  j = 1 j = 2 j = 3 j = 4
i = 2
  j = 1 j = 2 j = 3 j = 4
i = 3
  j = 1 j = 2 j = 3 j = 4
i = 4
  j = 1 j = 2 j = 3 j = 4
i = 5
  j = 1 j = 2 j = 3 j = 4

```

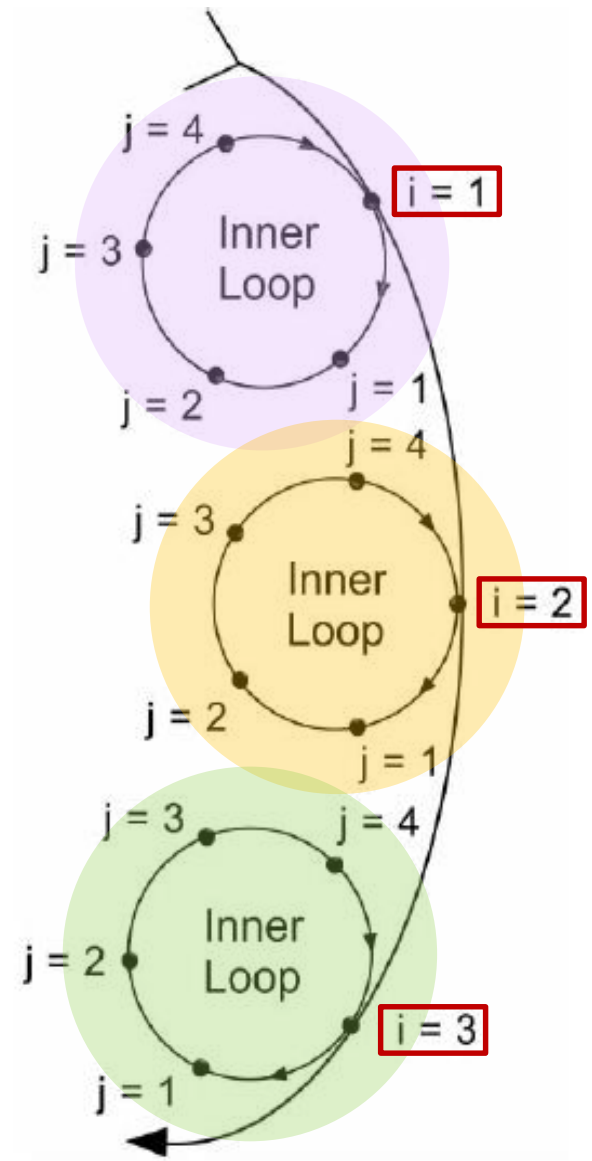


Figure 5.9 j loops once for each i

Example 1

```

05 SIZE = 5
06
07 for i in range(SIZE): # outer loop starts
08     for j in range(i + 1): # inner loop starts
09         print(j + 1, end=" ")
10     # inner loop ends
11     print("")
12 # outer loop end
...

```

Handwritten annotations:

- $i=0$ points to the first iteration of the outer loop.
- $0+1=1$ points to the first iteration of the inner loop.
- $0+1=1$ points to the second iteration of the inner loop.
- $0+1=1$ points to the third iteration of the inner loop.

```

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

```

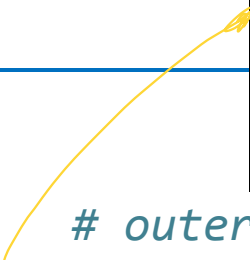
iteration	i	j	output
1	0	0	1
2	1	0 ... 1	1 2
3	2	0 ... 2	1 2 3
4	_____	_____	_____
5	_____	_____	_____

Example 2

```

05 SIZE = 5
06
07 for i in range(SIZE):           # outer loop starts
08     for j in range(SIZE - i):   # inner loop starts
09         print(j + 1, end=" ")
10                                     # inner loop ends
11     print("")
12                                     # outer loop end
...

```



```

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

```

iteration	i	j	output
1	0	0 ... 4	_____
2	1	_____	_____
3	2	_____	_____
4	3	_____	_____
5	4	_____	_____

Example 3

```

      *
     * *
    * * *
   * * * *
  * * * * *

```

```

05 SIZE = 5
06
07 for i in range(SIZE): → จำนวนบรรทัด
08     for j in range(SIZE - i): → จำนวน character
09         print(" ", end="")
10
11     for k in range(i + 1):
12         print("* ", end="")
13
14     print("")

```

iteration	i	j	k	output
1	0	_____	_____	_____
2	1	_____	_____	_____
3	2	_____	_____	_____
4	3	_____	_____	_____
5	4	_____	_____	_____

Basic Loop Programming Techniques

loop + if else

- Technique 1: Selection within a loop
- Technique 2: Input data validation
- Technique 3: Interactive loop control
- Technique 4: Evaluating equations

1: Selection within a Loop

```

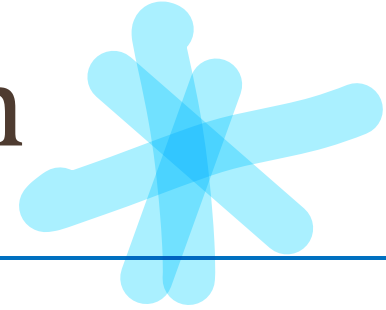
05 # this program computes the positive and negative sum
06 # of a set of MAXNUM user entered numbers
07
08 def count_pos_neg():
09
10     MAXNUM = 5
11     pos_total = 0
12     neg_total = 0
13     for i in range(MAXNUM):
14         x = float(input(""))
15         if x > 0:
16             pos_total += x
17         if x < 0:
18             neg_total += x
19
20     print("\nThe positive total is ", pos_total)
21     print("The negative total is ", neg_total)
...

```

3
 -2
 8
 -5
 1

The positive total is 12.0
 The negative total is -7.0

2: Input Data Validation



```
05 def verify_month_input():
06     while True:
07         month = int(input("Enter a month (1 - 12): "))
08         if 1 <= month <= 12:
09             return month
10         else:
11             print("Error - invalid month.")
12
...
```

3: Interactive Loop Control

```

05 # this program displays a table of numbers,
06 # their squares and cubes
07 # starting from the number 1.
08 # The final number in the table
09 # is input by the user
10
11
12 def display_sqare_and_cube():
13
14     final = int(input("Enter the final number: "))
15
16     print("Number Square Cube")
17     print("-----")
18
19     for num in range(1, final + 1):
20         print("%3d %7d %6d" % (num, num**2, num**3))
21

```

Enter the final number: 5
 Number Square Cube

```

-----
 1         1         1
 2         4         8
 3         9        27
 4        16        64
 5        25       125

```

4: Evaluating Equations

```

05 def equation1(start, end):
06
07     print("x value y value")
08     print("-----")
09     x = start
10     while x <= end:
11         y = 10 * (x ** 2) + 3 * x - 2
12         print("%4.2f %10.2f" % (x, y))
13         x += 0.5
14
15
16 equation1(4, 6)
17

```

x value	y value
-----	-----
4.00	170.00
4.50	214.00
5.00	263.00
5.50	317.00
6.00	376.00

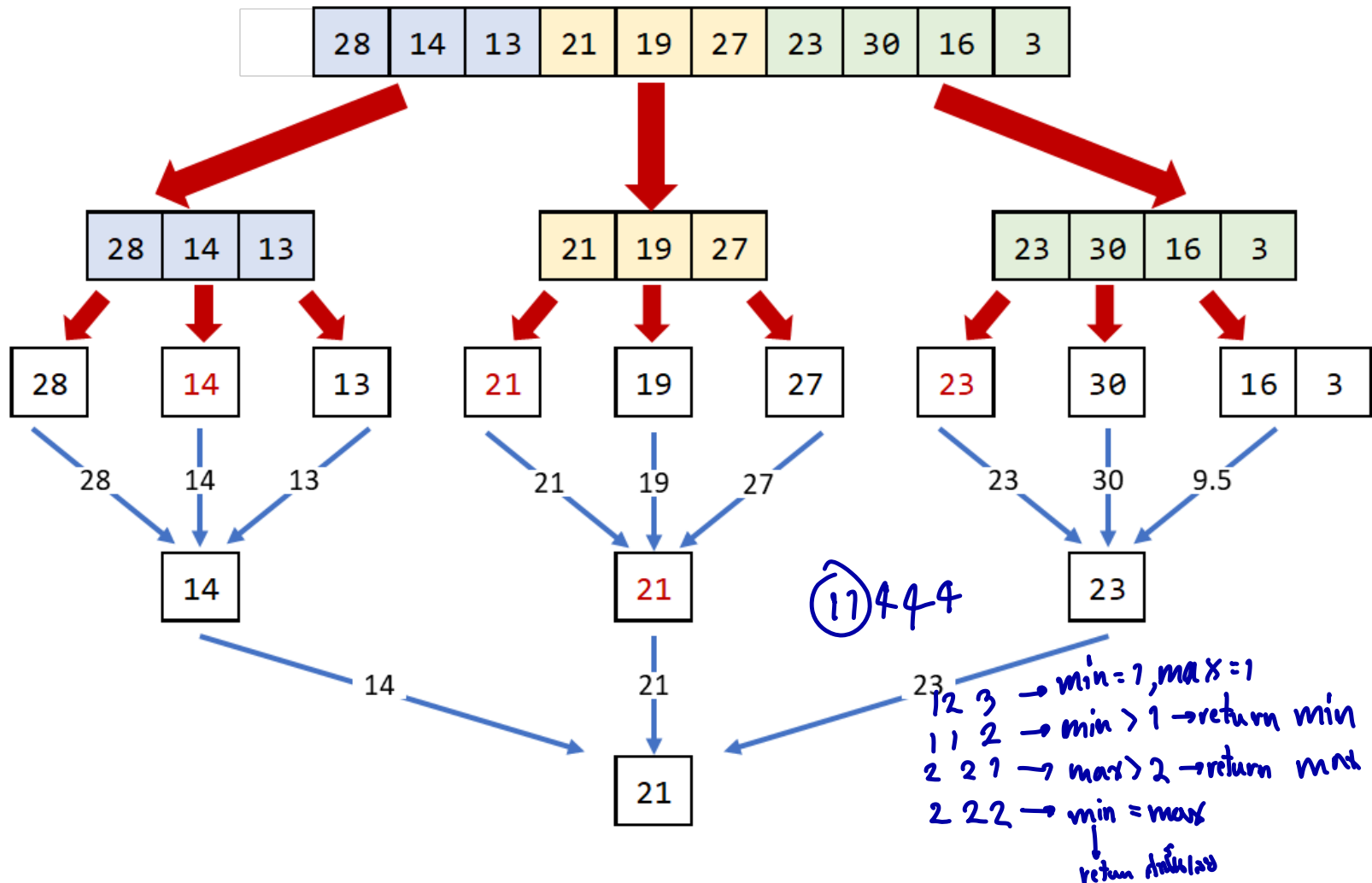
Common Programming Errors

- "Off by one" error, in which the loop executes either one too many or one too few times than intended
ทำเกินไป 1 Loop หรือ ทำขาดไป 1 Loop (ให้เช็คเงื่อนไขที่ใช้ในการควบคุม Loop ให้ดี)
- Using the assignment operator, =, instead of the equality operator, ==, in the tested expression
ใช้สัญลักษณ์สำหรับการเปรียบเทียบผิด คือ ใช้ = (สำหรับกำหนดค่าให้กับตัวแปร) แทนที่จะใช้ == (สำหรับเปรียบเทียบ)
- As with the if statement, repetition statements should not use the equality operator, ==, when testing single-precision or double-precision operands
ไม่ควรใช้การเปรียบเทียบในลักษณะ เท่ากับ == สำหรับการเปรียบเทียบตัวเลขทศนิยม เนื่องจากมีการปัดเศษ
 - พิจารณาการใช้ฟังก์ชันช่วยเปรียบเทียบเช่น `almost_equal()` หรือ `math.isclose()` แทน

References

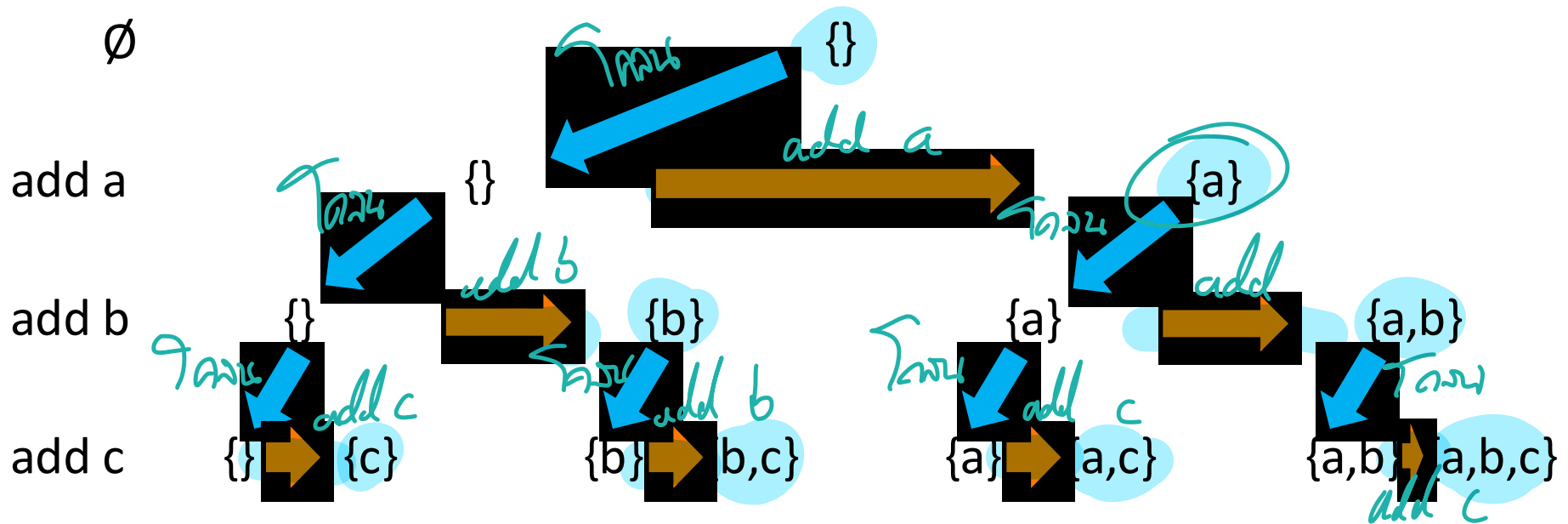
- <http://www.mathpages.com/home/kmath190.htm>
- <http://www.cs.cmu.edu/~112/notes/notes-monte-carlo.html>
- Gary J. Bronson *A First Book of ANSI C, Fourth Edition*
- Guttag, John V. *Introduction to Computation and Programming Using Python, Revised*

w12 Assignment – `median_of_median()`



w12 Assignment – subset_sum()

- For {a,b,c}



```
[set(), {'c'}, {'b'}, {'b', 'c'}, {'a'}, {'a', 'c'},
{'a', 'b'}, {'a', 'b', 'c'}]
```


PRACTICE

Practice 1

ให้เขียนฟังก์ชัน `rectangle1(n)` $n (n \geq 1)$ และแสดงผลดัง
output

```
Enter n: 5
1 2 3 4 5
  2 3 4 5
    3 4 5
      4 5
        5
```

- ตัวอย่างการ run 2

```
Enter n: 3
1 2 3
  2 3
    3
```

Practice 2

ให้เขียนฟังก์ชัน `rectagle2(n)` $n (n \geq 1)$ และแสดงผลดัง
output

```
Enter n: 5
      5
     4 5
    3 4 5
   2 3 4 5
  1 2 3 4 5
```

- ตัวอย่างการ run 2

```
Enter n: 3
      3
     2 3
    1 2 3
```

Practice 3

ให้เขียนฟังก์ชัน `rectangle3(n)` $n (n \geq 1)$ และแสดงผลดัง
output

```
Enter n: 5
5 4 3 2 1
 4 3 2 1
  3 2 1
   2 1
    1
```

- ตัวอย่างการ run 2

```
Enter n: 3
3 2 1
 2 1
  1
```

Practice 4

ให้เขียนโปรแกรมเพื่อรับค่าจำนวนเต็มทีประกอบด้วย 1 และ 0 ไม่เกิน 8 หลัก พร้อมแสดงผลให้อยู่ในรูป เลขฐาน 2 จำนวน 8 bit โดยให้ใส่ 0 ในจำนวนหลักที่ว่าง

- ตัวอย่างการ run 1

```
Input binary: 1101
|0000 1101| is 13 in decimal
```

- ตัวอย่างการ run 2

```
Input binary: 1102
Invalid input
Input binary: 11
|0000 0011| is 3 in decimal
```

Practice 5

ให้เขียนโปรแกรมเพื่อรับค่าจำนวนเต็ม (-, 0, +) และแสดงค่าจำนวนที่กลับหลักแล้ว ดังผลการ run ด้านล่าง

- ตัวอย่างการ run 1

```
Input an integer: 94857  
75849
```

- ตัวอย่างการ run 2

```
Input an integer: -1  
-1
```

Practice 6

ให้เขียนโปรแกรมเพื่อรับเลขจำนวนเต็ม x แล้วตรวจสอบว่า x เป็นจำนวนเฉพาะหรือไม่ ดังตัวอย่างการ run ด้านล่าง

- ตัวอย่างการ run 1

```
Input an integer: 1223
1223 is prime
```

- ตัวอย่างการ run 2

```
Input an integer: 1227
1227 is NOT prime
```

หมายเหตุ: จำนวนแรกทีหาร x ลงตัวหาก x ไม่ใช่จำนวนเฉพาะจะมีค่าไม่เกิน \sqrt{x}

Practice 7

ให้เขียนโปรแกรมเพื่อคำนวณเลข fraction ในฐาน 2 โดยรับค่าเฉพาะส่วนที่อยู่หลังจุด binary point ไม่เกิน 8 ตำแหน่ง พร้อมแสดงผลเป็นทศนิยม 8 ตำแหน่ง ดังผลการ run ด้านล่าง

```
Input binary fraction: 1101
0.1101 is the sum of
1/2 +
1/4 +
0/8 +
1/16
The result is 0.81250000
```


Practice 8

ให้นำข้อมูลประเภทจำนวนเต็ม 2 ค่า คือค่าแรก (first) และค่าสุดท้าย (last) ผ่านทาง keyboard และนับว่ามีจำนวนเฉพาะกี่จำนวน อะไรบ้าง

- ตัวอย่างการ run

```
Input first integer: 3
Input last integer: 20
Prime numbers between 3 and 20 are:
3 5 7 11 13 17 19
Total: 7
```

Practice 9

- **Problem Statement** ให้เขียนฟังก์ชัน `reverse(n)` เพื่อกลับหลักจำนวนเต็ม n และคืนค่าจำนวนที่กลับหลักแล้ว

<u>Input</u>	<u>Output</u>
13465	56431
0	0
-58	-85

Practice 10

- **Problem Statement** ให้เขียนฟังก์ชัน `crossed_number(n)` n ($n \geq 3$) เพื่อแสดงผลตัวเลขตั้งแต่ 1 - n ในรูปแบบดังแสดง

ตัวอย่างการ Run 1

```
Enter n: 6
1           6
  2       5
    3 4
    3 4
  2       5
1           6
```

ตัวอย่างการ Run 2

```
Enter n: 3
1  3
  2
1  3
```

Practice 11

- **Problem Statement** ให้เขียนฟังก์ชัน `square_pattern(n)` n ($n \geq 3$) เพื่อแสดงผลตัวเลขตั้งแต่ 1 - n ในรูปแบบดังแสดง

ตัวอย่างการ Run 1

```
Enter n: 5
1 2 3 4 5
2 2 3 4 5
3 3 3 4 5
4 4 4 4 5
5 5 5 5 5
```

ตัวอย่างการ Run 2

```
Enter n: 3
1 2 3
2 2 3
3 3 3
```

Practice 12

ให้นำข้อมูลประเภทจำนวนเต็ม 2 ค่า คือค่าแรก (first) และค่าสุดท้าย (last) ผ่านทาง keyboard และนับว่ามีจำนวนเฉพาะกี่จำนวน อะไรบ้างเช่น

7 ถึง 21: มี จำนวนเฉพาะ 5 ตัวคือ: 7, 11, 13, 17, 19

- ตัวอย่างการ run 1

```
Input first value: 7
Input last value: 21
The prime number(s) between 7 - 21 are
7 11 13 17 19
Total: 5 numbers
```

Practice 13

ตัวอย่าง Output

Average of Sale Amount of Week (1) = 12800.00
 Average of Sale Amount of Week (2) = 13050.00
 Average of Sale Amount of Week (3) = 13728.57
 Average of Sale Amount of Week (4) = 14500.00
 Average of Sale Amount for All = 13519.64

ให้เขียนโปรแกรมภาษา Python เพื่อหาค่าเฉลี่ยของยอดขายทั้งเดือน
 ค่าเฉลี่ยของยอดขายต่อสัปดาห์ ให้สอดคล้องกับตัวอย่างการ run
 โปรแกรมด้านล่าง โดยที่ค่าข้อมูลให้นำเข้าผ่านทาง keyboard

Week	Su	Mo	Tu	We	Th	Fr	Sa	Average (Per Week)
1	15000.00	10000.00	11500.00	9500.00	12600.00	14500.00	16500.00	12800.00
2	14500.00	11000.00	12500.00	9600.00	12650.00	14550.00	16550.00	13050.00
3	14800.00	11500.00	13500.00	9800.00	13500.00	15500.00	17500.00	13728.57
4	16000.00	12450.00	13550.00	10500.00	14500.00	16000.00	18500.00	14500.00
						Average (All)		13519.64

Practice 14

ให้นำข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
1
1 2
1 2 3
```

- ตัวอย่างการ run 2

```
Input row: 4
1
1 2
1 2 3
1 2 3 4
```

Practice 15

ให้นำข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
* * 1
* 2 2
3 3 3
```

- ตัวอย่างการ run 2

```
Input row: 4
* * * 1
* * 2 2
* 3 3 3
4 4 4 4
```


Practice 16

ให้นำข้อมูลจำนวนแถว (row) และ พิมพ์ผลลัพธ์ดัง
ตัวอย่างแสดงด้านล่าง

- ตัวอย่างการ run 1

```
Input row: 3
```

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

- ตัวอย่างการ run 2

```
Input row: 4
```

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