

# Exercises

By Nachai Paramesthanakorn

# Find day of year

## Input:

รับค่าวัน เดือน ปี เป็นจำนวนเต็มเก็บในตัวแปร d, m, y ตามลำดับ

กำหนดให้รับวัน เดือน ปี เป็นวันที่ที่ถูกต้อง

## Process:

หาว่าวัน เดือน ปี ที่รับเข้ามาเป็นวันที่เท่าไรของปี

กำหนดให้สร้างเป็นฟังก์ชัน

## Output:

แสดงผล วันที่เท่าไรของปีที่ได้จากการคำนวณ

## ตัวอย่าง

Input	Output
29 4 2024	120
9 1 2022	9
31 12 2024	366
31 12 2023	365

# Find day between date

## Input:

บรรทัดที่ 1 รับค่าวัน เดือน ปี เก็บในตัวแปร d1, m1, y1 คั่นด้วยช่องว่าง ตามลำดับ  
บรรทัดที่ 2 รับค่าวัน เดือน ปี เก็บในตัวแปร d2, m2, y2 คั่นด้วยช่องว่าง ตามลำดับ  
กำหนดให้รับวัน เดือน ปี เป็นวันที่ถูกต้อง

## Process:

หาว่าวันเดือนปีที่รับเข้ามาทั้ง 2 ชุดมีจำนวนวันต่างกัน กี่วัน  
กำหนดให้สร้างเป็นฟังก์ชัน

## Output:

แสดงผล จำนวนวันที่ต่างกัน (ไม่คิดเครื่องหมายลบ)

## ตัวอย่าง

Input	Output
29 4 2024 1 5 2024	2
31 3 2024 3 2 2024	57
12 1 2024 3 12 2024	326

# 4 Functions

```
1
2 """
3 จงเขียน 4 ฟังก์ชัน ให้ทำงานตามที่เขียนอธิบายกำกับแต่ละฟังก์ชัน ในโครงของโปรแกรมนี้
4 """
5
6 def make_int_list(x):
7     # รับสตริง x มาแยกและแปลงเป็น int เก็บไว้ใน list แล้ว return ผลลัพธ์
8     # เช่น x = '12 34 5' ได้ผลเป็น [12, 34, 5]
9     pass
10
11 def is_odd(e):
12     # คืนค่าจริงเมื่อ e เป็นจำนวนคี่ ถ้าไม่ใช่ คืนค่าเท็จ
13     pass
14
15 def odd_list(alist):
16     # คืน list ที่มีค่าเหมือน alist แต่มีเฉพาะตัวที่เป็นจำนวนคี่
17     # เช่น alist = [10, 11, 13, 24, 25] จะได้ [11, 13, 25]
18     pass
19
20 def sum_square(alist):
21     # คืนผลรวมของกำลังสองของแต่ละค่าใน alist
22     # เช่น alist = [1,3,4] ได้ผลเป็น (1*1 + 3*3 + 4*4) = 26
23     pass
24
25
26 print(make_int_list('1 2 3 4 5'))    # ต้องได้ [1, 2, 3, 4, 5]
27 print(is_odd(2222))                  # ต้องได้ False
28 print(odd_list([1,2,3,4,5,6,7]))     # ต้องได้ [1, 3, 5, 7]
29 print(sum_square([1,1,2,3]))         # ต้องได้ 15
```

# Calculate volume

จงเขียน ฟังก์ชันคำนวณหาผลลัพธ์

Input:

กำหนดให้ x คือ list ที่เก็บชุดข้อมูลขนาดของลูกบาศก์ (width, height, length, cube\_type)  
หลาย ๆ ชุด

กำหนดให้ y คือ dict เก็บค่าคงที่ a และ b ของแต่ละ cube\_type

Process:

1. หา ค่า a และ b ที่ต้องใช้จาก cube\_type
2. หาปริมาตรเฉลี่ย, ปริมาตรที่มากที่สุด, ปริมาตรที่น้อยที่สุด, ผลรวมของปริมาตรทั้งหมด และจำนวนข้อมูลทั้งหมด ของแต่ละ cube\_type
3. หาปริมาตรเฉลี่ย, ปริมาตรที่มากที่สุด, ปริมาตรที่น้อยที่สุด, ผลรวมของปริมาตรทั้งหมด และจำนวนข้อมูลทั้งหมด ของชุดข้อมูลทุก cube\_type ทั้งหมดใน x

วิธีการคำนวณหาปริมาตรของโปรแกรมมีดังนี้

1. ตรวจสอบ height ด้วยฟังก์ชัน check\_height() จะได้ค่า height และ s
2. ตรวจสอบ s ด้วยฟังก์ชัน check\_s() จะได้ค่า length
3. คำนวณปริมาตรจาก ฟังก์ชัน calculate\_volume()

สามารถเขียนฟังก์ชันเพิ่มเติมได้ (ถ้ามี)

จำนวนข้อมูล ให้แสดงผลลัพธ์เป็น int ที่เหลือให้แสดงทศนิยม 2 ตำแหน่ง

Output ที่ต้องการ:

Result:

A: average:525.01, max:1250.87, min:232.32, sum:4725.12 count:9

B: average:869.95, max:1875.98, min:63.36, sum:6959.56 count:8

C: average:232.61, max:471.11, min:42.24, sum:1628.3 count:7

S: average:513.82, max:2227.68, min:38.53, sum:3596.71 count:7

Total: average:545.47, max:2227.68, min:38.53, sum:16909.68  
count:31

# Calculate volume ต่อ

```
def check_height(height, width):  
    """มีการตรวจสอบดังนี้  
    ถ้า height เป็นเลขคู่ ให้เรียกใช้ ฟังก์ชัน get_new_height() แล้ว return ค่าที่ได้จาก get_new_height() และ s=2  
    ถ้า height เป็นเลขคี่ ให้ return height += 1 และ s=1  
  
    Args:  
        height (float): ความสูง  
        width (float): ความกว้าง  
  
    Returns:  
        float: height ใหม่ที่ได้จากการคำนวณ  
        int: ค่า s เป็น 1 หรือ 2  
    """  
    pass  
  
def get_new_height(height, width):  
    """คำนวณค่า height ใหม่ โดยใช้สูตร (height*b) + (width*a) + 0.5  
    โดยค่า a และ b เป็น global variable แล้ว return ค่า height  
  
    Args:  
        height (float): ความสูง  
        width (float): ความกว้าง  
  
    Returns:  
        float: height ใหม่ที่ได้จากการคำนวณ  
    """  
    global a, b  
    pass
```

# Calculate volume ต่อ

```
def check_s(s, length):
    """ตรวจสอบค่า s
    ถ้า s = 1 length ใหม่จะมีค่าเพียง 80% ของ (length + b)
    ถ้า s = 2 length ใหม่จะมีค่าเพียง 95% ของ (length - a)

    Args:
        s (int): 1 หรือ 2
        length (float): ความยาว
    Returns:
        float: length ใหม่ที่ได้จากการคำนวณ
    """
    global a, b
    pass

def calculate_volume(height, width, length):
    """คำนวณปริมาตรจาก height * width * length
    แล้ว return ผลลัพธ์

    Args:
        height (float): ความสูง
        width (float): ความกว้าง
        length (float): ความยาว
    Returns:
        float: ปริมาตร
    """
    pass
```

กำหนดให้

y = {

"A": (0.5, 0.3), # a=0.5, b=0.3

"B": (0.4, 0.4), # a=0.4, b=0.4

"C": (0.3, 0.2),

"S": (0.2, 0.1),

}

x = [[10, 15, 3, "A"], [22, 8, 6, "B"], [10, 7, 2, "C"], [11, 6, 4, "A"], [11, 4, 4, "A"], [15, 12, 2, "S"], [19, 16, 4, "S"], [11, 3, 1, "C"], [8, 5, 4, "B"], [9, 1, 4, "B"], [9, 11, 4, "C"], [12, 7, 5, "A"], [13, 8, 5, "S"], [16, 9, 9, "B"], [17, 12, 10, "B"], [15, 9, 2, "A"], [13, 8, 1, "S"], [14, 2, 6, "C"], [15, 4, 5, "A"], [12, 3, 4, "C"], [11, 7, 3, "A"], [16, 7, 7, "B"], [17, 17, 9, "S"], [15, 4, 6, "C"], [10, 8, 13, "B"], [13, 8, 5, "B"], [14, 6, 4, "S"], [9, 3, 2, "C"], [19, 7, 3, "S"], [21, 17, 3, "A"], [22, 6, 5, "A"],]