

[HW] Ch6_Conditions(if_else)_v01

1 จงเขียนฟังก์ชันส่งค่าคืนกลับเป็นวันในสัปดาห์ `get_day_week()` ตาม Flowchart ต่อไปนี้ โดยกำหนดให้อาร์กิวเมนต์วันเดือนปีที่ส่งให้ฟังก์ชันที่เป็นตัวเลข

เช่น ถ้าเป็น 21 พ.ค. 2027 ก็ให้กดคีย์เป็น

21 ↵

5 ↵

2027 ↵

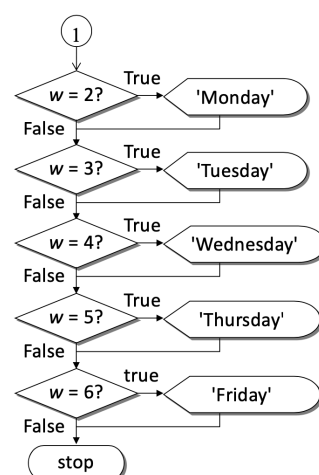
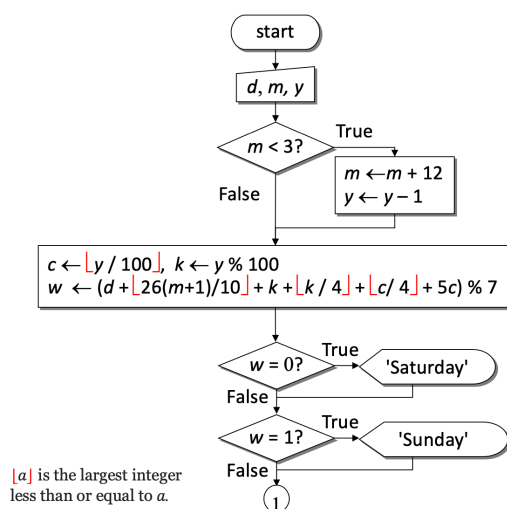
(↵ คือ ปุ่ม Enter (เอ็นเทอร์))

Case-1 - Output: (Keyboard Input: 2, 2, 2024)

Date = 2
Month Number = 2
Year (A.D.) = 2024
Friday

Case-2 - Output: (Keyboard Input: 10, 2, 2024)

Date = 10
Month Number = 2
Year (A.D.) = 2024
Saturday



https://en.wikipedia.org/wiki/Determination_of_the_day_of_the_week

In [1]:

```
# HW
# เขียนโค้ดด้านล่าง แล้วกด Shift+Enter
# By using the skeleton of code below.
```

```
def get_day_week(d, m, y):
```

```

day_of_week = ['Monday', 'Tuesday', 'Wednesday', \
               'Thursday', 'Friday', 'Saturday', 'Sunday']

### BEGIN SOLUTION
day_of_week = [ 'Saturday', 'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
pass

D=d
M=m
Y=y
if m<3:
    m=m+12
    y=y-1
c = int(y/100)
k = y%100

w=(d+int(26*(m+1)/10)+k+int(k/4)+int(c/4)+5*c)%7
if D==29 and M==2 and not((((Y % 4 == 0) and (Y % 100 != 0))) or (Y % 400 == 0)):
    return "Don't have this day."
else:
    return day_of_week[w]
### END SOLUTION

```

หลังจากเขียนโค้ดเสร็จแล้ว ให้ทำการทดสอบ ถ้าได้ผลลัพธ์ Pass/True ครบ 5 ครั้ง แสดงว่า Test ผ่านหมด

```

In [2]: # HW: For test-run
from datetime import datetime
from random import choice

print("@", datetime.now().strftime("%d/%m/%Y %H:%M:%S"))

test_N = 5

for i in range(test_N):
    # สุ่มโดยเช็คปีว่าเป็นปี Leap-year หรือไม่ และกำหนดจำนวนวันของแต่ละเดือน เพื่อไม่ให้วันที่เกิน
    years = list(range(1950, 2101)) # years 1950~2100
    y = choice(years)
    # Check for Leap-year
    if (((y % 4 == 0) and (y % 100 != 0))) or (y % 400 == 0):
        leap_flag = True
    else:
        leap_flag = False

    months = list(range(1, 13)) # 12 months list
    m = choice(months)
    if m in (1, 3, 5, 7, 8, 10, 12):
        m_length = 31
    elif m == 2:
        if leap_flag:
            m_length = 29
        else:
            m_length = 28
    else:
        m_length = 30

    dates = list(range(1, m_length))
    d = choice(dates)

```

```
print('{:>2}/{:>2}/{}'.format(d, m, y), end=' ')
ans = datetime(y,m,d).date().strftime('%A')
ans_flag = (get_day_week(d,m,y) == ans)
print('-> should be {:<10} : {}'.format(ans, ans_flag))
```

```
@ 25/02/2024 22:13:22
7/10/2098 -> should be Tuesday : True
9/ 4/2061 -> should be Saturday : True
17/ 1/2022 -> should be Monday : True
15/ 2/2026 -> should be Sunday : True
30/ 8/2026 -> should be Sunday : True
```

2 ค่าดัชนีมวลกาย (Body Mass Index) หรือเรียกย่อๆ ว่า BMI คือ ตัววัดมาตรฐานเพื่อประเมินสภาวะ
อ้วนหรือผอมในผู้ใหญ่ตั้งแต่อายุ 20 ปีขึ้นไป ซึ่งคำนวณได้จาก การใช้น้ำหนักตัว (\$w\$) เป็นกิโลกรัมและหาร
ด้วยส่วนสูง (\$h\$) ที่วัดเป็นเมตรยกกำลังสอง (Ref: <https://www.bangpakokhospital.com>) ตาม
สมการต่อไปนี้

$$BMI = \frac{w}{h^2}$$

จงเขียนโปรแกรมประเมินค่า BMI โดยให้แสดงผลการประเมินตามตาราง



ตัวอย่าง

Case-1 - Output: (Keyboard Input: 170, 62)

```
ส่วนสูง (cm)? 170
น้ำหนัก (kg)? 62
ผลการคำนวณ BMI อยู่ในเกณฑ์ สมส่วน
```

Case-2 - Output: (Keyboard Input: 165, 78)

```
ส่วนสูง (cm)? 165
น้ำหนัก (kg)? 78
ผลการคำนวณ BMI อยู่ในเกณฑ์ โรคอ้วน
```

In [9]: *# HW: BMI Evaluation*
By using the skeleton of code below.

```
def bmi_evaluate(w, h):  
    '''  
    คำนวณค่า BMI โดยน้ำหนัก (w) มีหน่วยเป็น kg และส่วนสูง (h) มีหน่วยเป็น cm  
    แล้วประเมินความสมดุลของน้ำหนักตัวต่อส่วนสูงว่าอยู่ในเกณฑ์ที่เหมาะสมหรือไม่  
    '''  
  
    bmi_value = w / (h/100)**2  
  
    ### BEGIN SOLUTION  
    if bmi_value>30:  
        return "โรคอ้วนอันตราย"  
    elif 29.9>=bmi_value>=25:  
        return "โรคอ้วน"  
    elif 24.9>=bmi_value>=23:  
        return "น้ำหนักเกิน"  
    elif 22.9>=bmi_value>=18.5:  
        return "สมส่วน"  
    elif bmi_value<18.5:  
        return "น้ำหนักต่ำกว่าเกณฑ์"  
    ### END SOLUTION  
  
    return result  
  
height = float(input("ส่วนสูง (cm)? "))  
weight = float(input("น้ำหนัก (kg)? "))  
  
print(f"ผลการคำนวณ BMI อยู่ในเกณฑ์ {bmi_evaluate(weight, height)}")
```

ผลการคำนวณ BMI อยู่ในเกณฑ์ สมส่วน

หลังจากเขียนโค้ดข้างต้นเสร็จ ให้รันโค้ดต่อไปนี้ ถ้าได้ผลลัพธ์ **Test Passed** ทั้งหมด 5 ครั้ง แสดงว่า Test ผ่านทุกเงื่อนไข

```
In [10]: # HW: For test-run  
from datetime import datetime  
print("@", datetime.now().strftime("%d/%m/%Y %H:%M:%S"))  
  
def test_bmi_evaluate(weight, height, expected_result):  
    result = bmi_evaluate(weight, height)  
  
    # Check if the result matches the expected result  
    if result == expected_result:  
        print(f"Test Passed: BMI for weight={weight} kg, height={height} cm is {  
    else:  
        print(f"Test Failed: BMI for weight={weight} kg, height={height} cm is {  
  
    # Test Cases  
    test_bmi_evaluate(45, 160, "น้ำหนักต่ำกว่าเกณฑ์")  
    test_bmi_evaluate(65, 170, "สมส่วน")  
    test_bmi_evaluate(75, 175, "น้ำหนักเกิน")  
    test_bmi_evaluate(85, 180, "โรคอ้วน")  
    test_bmi_evaluate(105, 185, "โรคอ้วน อันตราย")
```

@ 25/02/2024 22:24:28

Test Passed: BMI for weight=45 kg, height=160 cm is น้ำหนักต่ำกว่าเกณฑ์

Test Passed: BMI for weight=65 kg, height=170 cm is สมส่วน

Test Passed: BMI for weight=75 kg, height=175 cm is น้ำหนักเกิน

Test Passed: BMI for weight=85 kg, height=180 cm is โรคอ้วน

Test Failed: BMI for weight=105 kg, height=185 cm is โรคอ้วนอันตราย. Expected โรคอ้วน
อันตราย
