[HW] Ch8_Functions_v01

`1` Function `convert2fahrenheit()` to convert C to F

จงเขียนฟังก์ชันแปลงอุณหภูมิจาก `C` เป็น `F` `fahrenheit( )`

โดยมี `celsius` เป็นพารามิเตอร์ของอุณหภูมิที่รับมาในหน่วยองศาเซลเซียส (celsius; C) และคืนค่ากลับเป็นอุณหภูมีในหน่วยฟาเรนไฮต์ (Fahrenheit; F) (**เลขทศนิยม 2 ตำแหน่ง**) โดยใช้สูตรคำนวณดังนี้:

$$F = \frac{9}{5}C +32$$

โดยผู้ใช้จะต้องใส่อุณหภูมิในหน่วยองศาเซลเซียส, แล้วโปรแกรมจะคำนวณและแสดงผลลัพธ์ในหน่วยองศาฟาเรนไฮต์

ตัวอย่าง Output: (Keyboard Input: 99)

```
กรุณาใส่อุณหภูมิในหน่วยองศาเซลเซียส (C): 99
อุณหภูมิในหน่วยองศาฟาเรนไฮต์คือ  210.2 F
```

In [4]:
```python
# HW: Convert C to F
# เขียนโค้ดคำตอบลงเซลล์นี้ แล้วทำการรันโดยกด Shift+Enter

def convert2fahrenheit(celsius):  # นิยามฟังก์ชัน convert2fahrenheit ที่รับอุณหภูมินในห
    """ คืนค่ากลับเป็นอุณหภูมิองศาฟาเรนไฮต์ (F) """
    ### BEGIN SOLUTION
    # ใช้ฟังก์ชัน round() เพื่อให้ผลลัพธ์เป็นทศนิยม 2 ตำแหน่ง
    # return (celsius * 9 / 5 + 32) # 80% pass
    return round((celsius * 9 / 5) + 32, 2) # 100% pass คำนวณตามสูตรแล้วส่งค่าคืนกลั

    ## END SOLUTION


# รับข้อมูลอุณหภูมิในหน่วยเซลเซียสจากผู้ใช้
celcius = int(input('กรุณาใส่อุณหภูมิในหน่วยองศาเซลเซียส (C):'))
# เรียกใช้ฟังก์ชัน convert2fahrenheit และแสดงผลลัพธ์ที่ได้บนหน้าจอ
print('อุณหภูมิในหน่วยองศาฟาเรนไฮต์คือ ', convert2fahrenheit(celcius), 'F')
```

```
อุณหภูมิในหน่วยองศาฟาเรนไฮต์คือ  210.2 F
```

ทำการทดสอบโค้ดที่เขียนข้างต้น โดยรันเซลล์ต่อไปนี้ ถ้าได้ผลลัพธ์ `Test Passed` 3 ครั้ง ถือว่าผ่าน 100%

In [2]:
```python
# For test-run
# รันโค้ดนี้ต่อเพื่อตรวจสอบผลลัพธ์
from datetime import datetime

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

def test_convert2fahrenheit(celsius, expected_result):
    # result = round(convert2fahrenheit(celsius),2) # check the result with only
    result = convert2fahrenheit(celsius)
```

```
    # Check if the result matches the expected result
    if result == expected_result:
        print(f"Test Passed: {celsius} Celsius = {result} Fahrenheit")
    else:
        print(f"Test Failed: {celsius} Celsius = {result} Fahrenheit. Expected {

# Test Cases
test_convert2fahrenheit(0, 32) # 0 in Celsius
test_convert2fahrenheit(-273.15, -459.67)  # Absolute zero in Celsius
test_convert2fahrenheit(37, 98.6)  # Body temperature in Celsius
```

```
@ 06/03/2024 18:17:54
Test Passed: 0 Celsius = 32.0 Fahrenheit
Test Passed: -273.15 Celsius = -459.67 Fahrenheit
Test Passed: 37 Celsius = 98.6 Fahrenheit
```

In [3]:
```
# หน้าที่หลักของ Docstring
help(convert2fahrenheit)
```

```
Help on function convert2fahrenheit in module __main__:

convert2fahrenheit(celsius)
    คืนค่ากลับเป็นอุณหภูมิองศาฟาเรนไฮต์ (F)
```

---

2. การคำนวณค่าสถิติเบื้องต้น

จงเขียนโปรแกรมคำนวณค่าสถิติเบื้องต้นเพื่อคำนวณค่าเฉลี่ย (cal_mean), ค่าเบี่ยงแบน (cal_sd), ค่าสูงสุด (cal_max), และค่าต่ำสุด (cal_min) จากตัวเลขที่ผู้ใช้ป้อนผ่านทางคีย์บอร์ด แล้วแสดงผลลัพธ์ทางหน้าจอ โดยมีรายละเอียดดังนี้

1. ฟังก์ชัน cal_mean(*nums) ที่รับข้อมูลตัวเลขจำนวนมาก (ไม่จำกัดจำนวน) และคืนค่าเฉลี่ยของตัวเลขนั้นๆ, ถ้าไม่มีข้อมูลใด ๆ เข้ามา คืนค่ากลับเป็น None.
2. ฟังก์ชัน cal_sd(*nums) ที่รับข้อมูลตัวเลขจำนวนมาก (ไม่จำกัดจำนวน) และคืนค่าค่าเบี่ยงแบน (standard deviation) ของตัวเลข, ถ้าไม่มีข้อมูลใด ๆ เข้ามา คืนค่ากลับเป็น None.
3. ฟังก์ชัน cal_max(*nums) ที่รับข้อมูลตัวเลขจำนวนมาก (ไม่จำกัดจำนวน) และคืนค่าตัวเลขที่มีค่าสูงที่สุด, ถ้าไม่มีข้อมูลใด ๆ เข้ามา คืนค่ากลับเป็น None.
4. ฟังก์ชัน cal_min(*nums) ที่รับข้อมูลตัวเลขจำนวนมาก (ไม่จำกัดจำนวน) และคืนค่าตัวเลขที่มีค่าต่ำที่สุด, ถ้าไม่มีข้อมูลใด ๆ เข้ามา คืนค่ากลับเป็น None.
5. รับข้อมูลจากผู้ใช้ทางคีย์บอร์ดในรูปแบบตัวเลขที่แยกด้วยช่องว่าง โค้ดตัวอย่างเขียนไว้อยู่แล้ว สามารถแก้ไขได้หากจำเป็น
6. ทดสอบแต่ละฟังก์ชันและแสดงผลลัพธ์ โค้ดตัวอย่างเขียนไว้อยู่แล้ว สามารถแก้ไขได้หากจำเป็น

[Note]

สมการค่าเฉลี่ย (Mean):

$$\text{Mean} = \frac{1}{N} \sum_{i=1}^{N} x_i $$

สมการค่าเบี่ยงแบน (Standard Deviation):

$$\text{SD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \text{Mean})^2}$$

ตัวอย่าง Output-1: เลขจำนวนเต็ม (Keyboard Input: 1 2 3 4 5 6 7 8 9)

```
ป้อนตัวเลขที่ต้องการคำนวณ (แยกด้วยช่องว่าง):  1 2 3 4 5 6 7 8 9

ค่าเฉลี่ย: 5.0
ค่าเบี่ยงแบน: 2.581988897471611
ค่าสูงสุด: 9.0
ค่าต่ำสุด: 1.0
```

ตัวอย่าง Output-2: เลขจำนวนจริง (Keyboard Input: 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 9.9)

```
ป้อนตัวเลขที่ต้องการคำนวณ (แยกด้วยช่องว่าง):  1.1 2.2 3.3 4.4 5.5 6.6
7.7 8.8 9.9

ค่าเฉลี่ย: 5.5
ค่าเบี่ยงแบน: 2.8401877872187726
ค่าสูงสุด: 9.9
ค่าต่ำสุด: 1.1
```

In [13]:

```python
# HW: Mean (Average) Function
# เขียนโค้ดคำตอบลงเซลล์นี้ แล้วทำการรันโดยกด Shift+Enter

# ประกาศฟังก์ชัน โดยมี arguments เป็นแบบ packed arguments *args เพื่อรับค่าไม่จำกัดจำนวน
def cal_mean(*nums):
    if len(nums) > 0:
        mean = (1/len(nums)) * sum(nums)
    else : return None
    return mean
# ฟังก์ชันคำนวณค่าเบี่ยงแบน (Standard Deviation)
def cal_sd(*nums):
    sd1 = 0
    if len(nums) > 0 :
        mean = (1/len(nums)) * sum(nums)
        for i in range(len(nums)) :
            sd1 += (nums[i]-mean)**2
            sd2 = (1/len(nums))*sd1
            sd = sd2**(1/2)
    else : return None
    return sd
# ฟังก์ชันคำนวณค่าสูงสุด
def cal_max(*nums):
    if len(nums) > 0:
        max = nums[0]
        for i in range(len(nums)):
            if max < nums[i]:
                max = nums[i]
    else : return None
    return max
# ฟังก์ชันคำนวณค่าต่ำสุด
def cal_min(*nums):
    if len(nums) > 0:
        min = nums[0]
        for i in range(len(nums)):
            if nums[i] <= min:
                min = nums[i]
```

```
    else : return None
    return min
inp = input("ป้อนตัวเลขที่ต้องการคำนวณ (แยกด้วยช่องว่าง): ") # รับข้อมูลตัวเลขจากผู้ใช้ (อยู่ในรู
num_strs = inp.split() # แยกข้อมูลตัวเลขออกจาก string ด้วย space (เก็บอยู่ในรูป list)

# แปลง string ให้เป็นตัวเลข (float) และเก็บในลิสต์ nums
nums = []
for num_str in num_strs:
    nums.append(float(num_str))


# เรียกใช้ฟังก์ชันแต่ละฟังก์ชันและแสดงผลลัพธ์
print(f"ค่าเฉลี่ย: {cal_mean(*nums)}")
print(f"ค่าเบี่ยงแบน: {cal_sd(*nums)}")
print(f"ค่าสูงสุด: {cal_max(*nums)}")
print(f"ค่าต่ำสุด: {cal_min(*nums)}")
```

```
ค่าเฉลี่ย: 5.0
ค่าเบี่ยงแบน: 2.581988897471611
ค่าสูงสุด: 9.0
ค่าต่ำสุด: 1.0
```

หลังจากที่เขียนเสร็จแล้ว ให้ทำการรันเซลล์ถัดไปเพื่อตรวจสอบว่าผ่าน (Passed) ทั้งหมดกี่ครั้ง และไม่ผ่าน (Failed) ทั้งหมดกี่ครั้ง

ทดสอบ 3 รายการ ถ้า Passed $3*4 = 12$ ครั้ง --> ถือว่าผ่าน 100%

In [14]:
```
# For test-run
# รันโค้ดต่อไปนี้ ตรวจสอบว่าผ่าน (Passed) ทั้งหมดกี่รายการ และไม่ผ่าน (Failed) ทั้งหมดกี่รายการ
# ทดสอบและเปรียบเทียบผลลัพธ์ของฟังก์ชันที่เขียนกับผลลัพธ์จาก numpy ในกรณีข้อมูลทดสอบต่าง ๆ
import numpy as np
from datetime import datetime

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

def test_functions(sample_data):
    # ทดสอบและเปรียบเทียบกับ numpy

    if len(sample_data) > 0:
        mean_result = cal_mean(*sample_data)
        mean_numpy = np.mean(sample_data)
        print(f"Mean Test: {'Passed' if abs(mean_result - mean_numpy) < 1e-6 els

        sd_result = cal_sd(*sample_data)
        sd_numpy = np.std(sample_data)
        print(f"Standard Deviation Test: {'Passed' if abs(sd_result - sd_numpy)

        max_result = cal_max(*sample_data)
        max_numpy = np.max(sample_data)
        print(f"Max Test: {'Passed' if max_result == max_numpy else 'Failed'}")

        min_result = cal_min(*sample_data)
        min_numpy = np.min(sample_data)
        print(f"Min Test: {'Passed' if min_result == min_numpy else 'Failed'}")
    else:
        mean_result = cal_mean(*sample_data)
        print(f"Mean Test: {'Passed' if mean_result == None else 'Failed'}")

        sd_result = cal_sd(*sample_data)
        print(f"Standard Deviation Test: {'Passed' if sd_result == None else 'Fa
```

```
        max_result = cal_max(*sample_data)
        print(f"Max Test: {'Passed' if max_result == None else 'Failed'}")

        min_result = cal_min(*sample_data)
        print(f"Min Test: {'Passed' if min_result == None else 'Failed'}")


# ข้อมูลทดสอบ Int, Float
sample_data0 = []
sample_data1 = [45, 67, 87, 21, 43, 98, 28, 23, 28, 75]
sample_data2 = [-1.2246467991473532e-14, -64.27876096865394, -98.48077530122082,


# ทดสอบและเปรียบเทียบกับ numpy สำหรับ sample_data1
print(f"\nTesting for Sample Data 0: {sample_data0}")
test_functions(sample_data0)

# ทดสอบและเปรียบเทียบกับ numpy สำหรับ sample_data1
print(f"\nTesting for Sample Data 1: {sample_data1}")
test_functions(sample_data1)

# ทดสอบและเปรียบเทียบกับ numpy สำหรับ sample_data2
print(f"\nTesting for Sample Data 2: {sample_data2}")
test_functions(sample_data2)
```

```
@ 12/03/2024 20:13:35

Testing for Sample Data 0: []
Mean Test: Passed
Standard Deviation Test: Passed
Max Test: Passed
Min Test: Passed

Testing for Sample Data 1: [45, 67, 87, 21, 43, 98, 28, 23, 28, 75]
Mean Test: Passed
Standard Deviation Test: Passed
Max Test: Passed
Min Test: Passed

Testing for Sample Data 2: [-1.2246467991473532e-14, -64.27876096865394, -98.4807
7530122082, -86.60254037844388, -34.20201433256688, 34.20201433256688, 86.6025403
7844383, 98.48077530122082, 64.27876096865394, 1.2246467991473532e-14]
Mean Test: Passed
Standard Deviation Test: Passed
Max Test: Passed
Min Test: Passed
```

---

3 โปรแกรมลำดับเลขฟีโบนักซี (Fibonacci numbers) โดยใช้การเรียกซ้ำ (Recursion)

เลขฟีโบนักซีสามารถเขียนเป็นอนุกรมได้ดังนี้คือ

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, F_{n-2}, F_{n-1}, F_n, ...

โดยแต่ละเลขเป็นผลรวมของเลขสองตัวก่อนหน้า กล่าวคือ ตัวเลขลำดับที่ n (F_n) จะเท่ากับตัวเลขลำดับที่ n-1 (F_{n-1}) บวกกับตัวเลขลำดับที่ n-2 (F_{n-2}) หรือเขียนเป็นสมการได้ดังนี้

$F_n = F_{n-1}+F_{n-2}$ (โดยที่ $F_0=0,F_1=1$)

ตัวอย่างเลขฟีโบนักชีลำดับที่ 0 ถึงลำดับที่ 20 แสดงเป็นตารางได้ดังนี้

F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18
0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	258

จงเขียนฟังก์ชัน `fibo_n(n))` ที่ใช้การเรียกซ้ำ (recursion) เพื่อคำนวณหาเลขฟีโบนักชีลำดับที่ n ตามสูตร $F(0)=0,F(1)=1$ และ $F(n) = F{n-1}+F{n-2}$ (สำหรับ n > 1)

Ref: Wiki,MathIsFun

Note: Recursion เป็นแนวคิดที่พบได้ทั่วไปทางคณิตศาสตร์และการเขียนโปรแกรม หมายถึงฟังก์ชันที่เรียกตัวเอง (call itself)

ตัวอย่าง Output: (สามารถตรวจสอบว่าได้ผลลัพธ์เหมือนแสดงในตารางหรือไม่)

```
F(0): 0
F(1): 1
F(2): 1
F(3): 2
...
...
F(19): 4181
F(20): 6765
```

In [10]:
```python
# HW: Fibonacci sequence ลำดับฟีโบนักชีโดยใช้การเรียกซ้ำ (Recursion)
# เขียนโค้ดคำตอบลงเซลล์นี้ แล้วทำการรันโดยกด Shift+Enter

def fibo_n(n):
    """
    Function: Fibonacci numbers (Fn): F0=0, F1=1, Fn=Fn-1+Fn-2
    Objective: To compute the value of Fn.
    Input: nth
    Output: nth Fibonacci number
    """
    ### BEGIN SOLUTION
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else :
        return fibo_n(n-1) + fibo_n(n-2)
    ### END SOLUTION


# ลองแสดงเลขฟีโบนักชีลำดับที่ 0 ถึงลำดับที่ 20 ว่าได้ผลลัพธ์เช่นเดียวกับในตารางหรือไม่
for i in range(21):
       print('F({}): {}'.format(i, fibo_n(i)) )
```

```
F(0): 0
F(1): 1
F(2): 1
F(3): 2
F(4): 3
F(5): 5
F(6): 8
F(7): 13
F(8): 21
F(9): 34
F(10): 55
F(11): 89
F(12): 144
F(13): 233
F(14): 377
F(15): 610
F(16): 987
F(17): 1597
F(18): 2584
F(19): 4181
F(20): 6765
```

ทดสอบโค้ดที่เขียนข้างต้น โดยรันเซลล์ต่อไปนี้ ถ้าได้ผลลัพธ์ `Test Passed` 4 ครั้ง ถือว่าผ่าน 100%

In [11]:
```python
# For test-run
# รันโค้ดนี้ต่อเพื่อแสดงผล print และตรวจสอบผลลัพธ์
from datetime import datetime
print("@", datetime.now().strftime("%d/%m/%Y %H:%M:%S"))

def test_fibo_n(n, expected_result):
    result = fibo_n(n)

    # Check if the result matches the expected result
    if result == expected_result:
        print(f"Test Passed: F({n}) = {result}")
    else:
        print(f"Test Failed: F({n}) = {result}. Expected {expected_result}")

# Test Cases
test_fibo_n(0, 0) # F(0) = 0
test_fibo_n(1, 1) # F(1) = 1
test_fibo_n(25, 75025) #F(25): 75025
golden = (1 + 5 ** 0.5) / 2
print(f"Test Passed: Golden ratio" if round(fibo_n(10)/fibo_n(9), 2) == round(go
```

```
@ 12/03/2024 20:06:22
Test Passed: F(0) = 0
Test Passed: F(1) = 1
Test Passed: F(25) = 75025
Test Passed: Golden ratio
```