ข้อ 1.

เรียกใช้ library math เพื่อช่วยในการคำนวน

```
==================== RESTART: C:\Users\KHUK\python\hw_01.py ====================
#*##*##*##*##*##*##*##*##*##*#
#*#   Quartdatic Solving   #*#
#*##*##*##*##*##*##*##*##*##*#
input a=5
input b=6
input c=1
#*##*##*##*##*##*##*##*##*##*#
x =-0.2 or x=-1.0
>>>
```

```
from math import *  ;''' เรียกใช้ libraly math '''
def functionQ(a,b,c): ;''' สร้าง function '''
    x1 = (-b + sqrt((b**2)-(4*a*c)))/(2*a)
    x2 = (-b - sqrt((b**2)-(4*a*c)))/(2*a)
    return x1,x2 ;''' ส่งค่าที่คำนวนแล้วกลับ ''''

print("#*#"*10) ;''' แสดง #*# 10 รอบ '''
print("#*#   Quartdatic Solving   #*#")
print("#*#"*10)
a = float(input("input a=")) ;''' รับค่า float จากแป้นพิมพ์ '''
b = float(input("input b="))
c = float(input("input c="))
print("#*#"*10)
x1,x2 = functionQ(a,b,c) ;''' นำค่าที่กรอกเข้ามา ไปคำนวนใน function '''
print("x ={x1} or x={x2}".format(x1=x1,x2=x2))
```

2.

ใช้ library math ในการช่วยคำนวณ

โดยเช็คราคากับเงินที่มีก่อนเพื่อแยกการทำงาน

ถ้าเงินน้อยกว่าราคา Nintendo ให้บอกจำนวนเงินที่ต้องการเพิ่ม

ถ้าเงินมากกว่าให้คำนวณว่าซื้อได้กี่เครื่อง และขาดอีกเท่าไหร่ถึงจะได้อีกเครื่อง

```
Price of nintendo switch :1000
Your money :1500
คุณสามารถซื้อได้ 1
คุณขาดเงินอีก 500.0 จะสามารถซื้อเพิ่มได้อีกเครื่อง
>>>
```

```
Price of nintendo switch :1000
Your money :700
Not enought money
Need more 300.0
>>>
```

```
from math import * ;''' เรียกใช้ library math  '''
nintendo = float(input("Price of nintendo switch :"))   ;''' รับค่าราคาเครื่อง nintendo '''
money = float(input("Your money :"))   ;''' รับค่าจำนวนเงินที่มี  '''
if(nintendo>money):   ;''' เงื่อนไขถ้าราคา nintendo มากกว่า เงินที่มี  '''
    print("Not enought money")   ;''' แสดงข้อความ '''
    print("Need more {x}".format(x=nintendo-money))  ;''' แสดงข้อความบอกว่าขาดอีกเท่าไหร่ถึงจะซื้อได้ '''

else:
    amount = floor(money / nintendo)  ;''' หาจำนวนเครื่องที่สามารถซื้อได้ '''
    print("คุณสามารถซื้อได้ {x}".format(x=amount))  ;'''แสดงข้อความ  '''
    money=((amount+1)*nintendo)-money  ;''' คำนวนว่าต้องการอีกเท่าไหร่ถึงจะได้อีกเครื่อง '''
    print("คุณขาดเงินอีก {x} จะสามารถซื้อเพิ่มได้อีกเครื่อง".format(x=money))  ;''' แสดงข้อความ '''
```

3.

ใช้ library math มาช่วยในการคำนวณ

โดยนำ ชม. ที่ได้มาคำนวณหาแคลอรี่

แล้วนำแคลอรี่รวม มาหาว่าสามารถลดได้ถึง 1 ปอนด์ไหม

ถ้าลดได้ให้คำนวณว่าใช้กี่วัน

ถ้าไม่ได้ให้แจ้งผิดพลาด

```
---------- RESTART: C:\
Enter Target (pound) : 5
Enter hour of Cycling : 8
Enter hour of Jogging : 6
Enter hour of Swimming : 2
Used 4 days
>>>
```

```
Enter Target (pound) : 5
Enter hour of Cycling : 2
Enter hour of Jogging : 2
Enter hour of Swimming : 2
Fail!
```

```
from math import *  ;''' เรียกใช้ library math '''
target = float(input("Enter Target (pound) : "))  ;''' รับค่าเป้าหมายที่ต้องการลด '''
cycling = float(input("Enter hour of Cycling : "))  ;''' รับค่าชั่วโมงที่ปั่นจักรยาน '''
jogging = float(input("Enter hour of Jogging : "))  ;''' รับค่าชั่วโมงที่วิ่ง '''
swimming = float(input("Enter hour of Swimming : "))  ;''' รับค่าชั่วโมงที่ว่ายน้ำ '''
cycling = cycling * 200  ;''' นำจำนวนชั่วโมงมาหาแคลอรี่ '''
jogging = jogging * 475 ;''' นำจำนวนชั่วโมงมาหาแคลอรี่ '''
swimming = swimming * 275 ;''' นำจำนวนชั่วโมงมาหาแคลอรี่ '''

pound = (cycling + jogging + swimming)/3500 ;''' นำแคลอรีมารวมกันและหารเพื่อหาค่าน้ำหนัก (ปอนด์) '''
day = target / pound ;''' นำเป้าหมายกับปริมาณที่ลดได้ต่อวันมาหารเพื่อคำนวนวันที่ใช้ลดน้ำหนัก '''
day = floor(day)+1 ;''' ปัดเศษขึ้น '''

if(pound<1): ;''' ถ้าปริมาณที่ลดต่อวันน้อยกว่า 1 ปอนด์ ถือว่าไม่สำเร็จ '''
    print("Fail!") ;''' แสดงข้อความ '''
else:
    print("Used {x} days".format(x=day)) ;''' แสดงข้อความว่าต้องใช้เวลากี่วันถึงจะลดได้ตามเป้า '''
```

4.

ใช้การคำนวณแบบวนลูปเพื่อหาค่า factorial

โดยให้ใส่เลข 1 จำนวน และสร้างลูปโดยสุดเลขตัวนั้น

ทำการคูณทบ ไปตามรอบ เพื่อหาค่า

```
Cal Factorial
number :5
resul =  120
Cal Factorial
number :6
resul =  720
>>>
```

```
sum = 1 ;''' สร้างค่าเริ่มต้น '''
print("Cal Factorial") ;''' แสดงข้อความ '''
inte = int(input("number :")) ;''' รับตัวเลขที่ต้องการค่า '''
for i in range(1, inte+1): ;''' สร้าง loop โดยวนรอบตั้งแต่ 1 ถึง ค่าที่รับมา '''
    sum = sum * i ;''' นำค่าที่รับมาคูณ '''

print("resul = ", sum) ;''' แสดงผลลัพธ์ '''
```

5.

รับค่าเงินต้นมา และเรียกใช้ฟังก์ชันคำนวณ

โดยมีการใช้เงื่อนไขในการแบ่งเพื่อหา ตัวคูณภาษีกับเงินภาษีคงที่ในวงเงินนั้น ๆ

โดยหลังจากแบ่งได้แล้ว นำเงินต้นมาคำนวนกับตัวคูณภาษี เพื่อหาภาษีก่อน

แล้วนำไปรวมกับ เงินต้นกับเงินภาษีคงที่

เพื่อที่จะได้ยอดรวม

```
Enter your income :350000
Your cost (including tax) : 392500.0
>>>
```

```
Enter your income :2540000
Your cost (including tax) : 3667000.0
```

```python
def taxes(income):  ;''' สร้าง function และรับพารามิเตอร์ 1 ตัว '''
    if(income >=1 and income <=150000): ;''' เช็คค่าที่รับเข้ามา  '''
        tax = 1  ;''' ตัวคูณภาษี '''
        net = 0 ;''' ค่าภาษีคงที่ '''
    elif(income<=300000):
        tax = 5 ;''' ตัวคูณภาษี '''
        net = 0 ;''' ค่าภาษีคงที่ '''
    elif(income<=500000):
        tax = 10 ;''' ตัวคูณภาษี '''
        net = 7500 ;''' ค่าภาษีคงที่ '''
    elif(income<=750000):
        tax = 15 ;''' ตัวคูณภาษี '''
        net = 27500 ;''' ค่าภาษีคงที่ '''
    elif(income<=1000000):
        tax = 20 ;''' ตัวคูณภาษี '''
        net = 65000 ;''' ค่าภาษีคงที่ '''
    elif(income<=2000000):
        tax = 25 ;''' ตัวคูณภาษี '''
        net = 115000 ;''' ค่าภาษีคงที่ '''
    elif(income<=4000000):
        tax = 30 ;''' ตัวคูณภาษี '''
        net = 365000 ;''' ค่าภาษีคงที่ '''
    elif(income>4000000):
        tax = 35 ;''' ตัวคูณภาษี '''
        net = 965000 ;''' ค่าภาษีคงที่ '''

    tax = ((income *tax/100 )+ net)+income ;'''นำเงินต้นมาคำนวนหาภาษีแล้วบวกด้วยค่าคงที่บวกเงินต้น '''

    return tax ;''' ส่งค่าที่คำนวนได้คืน '''

income = int(input("Enter your income :")) ;''' รับค่าเงินที่ต้องการคำนวน '''
print("Your cost (including tax) : {x}".format(x=taxes(income))) ;''' แสดงยอดภาษีรวมเงินต้น '''
```