MA1008 Introduction to Computational Thinking
Week 8 Tutorial: Functions
The first six questions are for discussion in class. Students should attempt the rest in their own time.

1. List the parts in a function definition and explain the purpose of each part.

2. Give three reasons why functions are useful.

3. What does this function do?

```
def Func(num):
    total = 0
    while num > 0:
        total = total + num*(num-1)
        num = num -1
    return total
```

4. What is a function call? How do you call a function? Given the function in Q3, what are the values of `x` in the following statements?
   i. `x = Func(5)`
   ii. `x = Func(5.5)`
   iii. `x = Func('5')`
   iv. `x = Func()`

5. What does the following code print? Explain.

```
number = 50
def Func(number):
   print(number)
   number=2
   print(number)

Func(number)
print(number)
```

6. Examine the following code and predict the outputs.

```
confusing = 0
def do_work(num):
   confusing = -50
   confusing += num
   print("confusing in do_work is ", confusing)
   return confusing

confusing = 50
print("confusing in main is ", confusing)
confusing += do_work(confusing)
print("confusing in main is ", confusing)
confusing += do_work(confusing)
print("confusing in main is ", confusing)
confusing += do_work(confusing)
```

The same code is in the file confusing.py in the hands-on folder. Copy it into the working folder and run it. Are your predictions the same as the program output?

7. A year is a *leap year* if it is divisible by 4 and not by 100, or if it is divisible by 400. Otherwise, it is not a leap year. Write a function that takes in an integer value and returns true if the value represents a leap year and false otherwise.

8. Define a 2D vector as a 2-tuple. Write a function that takes two vectors v1 and v2 as inputs and returns their sum.