ECC102 – PROGRAMMING AND PROBLEM-SOLVING
MIDTERM EXAM WORKSHEET
EXAMPLE QUESTIONS 1

1. In economics, the percentage rate of inflation for a period of time is calculated based on the final value F of a commodity and the initial value I of the commodity, using the formula $\left(\frac{F-I}{I}\right) * 100$. Write a Python function *inflation(initial, final)* to compute and return the inflation rate given the initial and final values of a commodity.

2. Consider the following Python program where **m and n** are assumed to be positive integers:

```
def mystery(n, m):
    p = 0
    e = 0
    while e < m:
        p = p + n
        e = e + 1
    return p

p = mystery(4,3)
```

Trace this program, *showing the value of e and p* in the table above *at the end of each loop iteration*.

p	e
0	0
---	---
---	---
---	---
---	---

3. Suppose that the return statement was indented as below. What would **mystery(4, 3)** return in this case?

```
def mystery(n, m):
    p = 0
    e = 0
    while e < m:
        p = p + n
        e = e + 1
        return p

p = mystery(4,3)
```

p = ……..

4. Consider the following Python program where **"sorted"** is used to sort split strings alphabetically:

```
def print_string(s, title=False, sort=False):
    if title:
        s = s.title()
    if sort:
        s = ' '.join(sorted(s.split()))

    print (s)

print_string('python is the best')
print_string('python is the best', sort=True)
print_string('python is the best', True, True)
```

Trace this program and determine the output.

5. Consider the following Python program:

```
numbers = [5, 4, 7, 0, 1]
count = 0

for number in numbers:
    if number:
        break
    count += 1

print(count)
```

a. Trace the program and determine the output.

b. Convert the following Python code to **use a while loop instead of a for loop**.

6. Write a Python code to determine if the given list of numbers **is sorted in ascending order.** You should use **a user-defined function "def is_sorted(numbers):"**. It should print **"sorted"** or **"unsorted".** For example, **numbers = [5, 6, 11, 8, 10]** is **unsorted**, while **numbers = [5, 6, 7, 8, 10]** is **sorted**.
Note that the use of built-in functions is not allowed!

7. Write a Python code to determine the priorities of the arithmetic operators (+, -, *, /). You should use a user-defined function named **"test_higher_priority"** which accepts **two operators as arguments** and determines the priority for those operators. You have to **define the operators and priorities in a dictionary**. When the function is called in the main program by passing the following arguments, the program should respond as shown below.

Calling the function	Expected Output
test_higher_priority('*','-')	# The priority of * is higher than -.
test_higher_priority('+','-')	# The priority of + and - is equal.
test_higher_priority('+','*')	# The priority of * is higher than +.
test_higher_priority('+','/')	# The priority of / is higher than +.
test_higher_priority('*','/')	# The priority of * and / is equal.

Note that the use of built-in functions is not allowed!