



National University of Modern Languages

Artificial Intelligence - LAB

Lab # 5

BSSE 5(M)

Submitted By:

Muhammad Umair (12093)

Submitted To:

Sir Faiq

Question 1

Objectives:

- becoming familiar with the `print()` function and its formatting capabilities;
- practicing coding strings;
- experimenting with Python code.

Scenario

Write a one-line piece of code, using the `print()` function, as well as the newline and escape characters, to match the expected result outputted on three lines.

Expected output

```
"I'm"  
""learning""  
""""Python""""
```

Question 2

Objectives

Familiarize the student with:

- using the `if-elif-else` statement;
- finding the proper implementation of verbally defined rules;
- testing code using sample input and output.

Scenario

As you surely know, due to some astronomical reasons, years may be *leap* or *common*. The former are 366 days long, while the latter are 365 days long.

Since the introduction of the Gregorian calendar (in 1582), the following rule is used to determine the kind of year:

- if the year number isn't divisible by four, it's a *common year*;
- otherwise, if the year number isn't divisible by 100, it's a *leap year*;
- otherwise, if the year number isn't divisible by 400, it's a *common year*;
- otherwise, it's a *leap year*.

Look at the code in the editor – it only reads a year number, and needs to be completed with the instructions implementing the test we've just described.

The code should output one of two possible messages, which are Leap year or Common year, depending on the value entered.

It would be good to verify if the entered year falls into the Gregorian era, and output a warning otherwise: Not within the Gregorian calendar period. Tip: use the `!=` and `%` operators.

Test your code using the data we've provided.

Test Data

Sample input: 2000

Expected output: Leap year

Sample input: 2015

Expected output: Common year

Sample input: 1999

Expected output: Common year

Sample input: 1996

Expected output: Leap year

Sample input: 1580

Expected output: Not within the Gregorian calendar period

Question 3

Objectives

- improving the student's skills in defining, using and testing functions.

Scenario

A car's fuel consumption may be expressed in many different ways. For example, in Europe, it is shown as the amount of fuel consumed per 100 kilometers. In the USA, it is shown as the number of miles traveled by a car using one gallon of fuel.

Your task is to write a pair of functions converting l/100km into mpg, and vice versa.

The functions:

- are named l100kmtompg and mpgtol100km respectively;
- take one argument (the value corresponding to their names)

Complete the code in the editor.

Run your code and check whether your output is the same as ours.

Here is some information to help you:

- 1 American mile = 1609.344 metres;
- 1 American gallon = 3.785411784 litres.

Test Data

```
print(l100kmtompg(3.9))
```

```
print(l100kmtompg(7.5))
```

```
print(l100kmtompg(10.))
```

```
print(mpgtol100km(60.3))
```

```
print(mpgtol100km(31.4))
```

```
print(mpgtol100km(23.5))
```

Expected output:

```
60.31143162393162
```

```
31.361944444444444
```

```
23.521458333333333
```

```
3.9007393587617467
```

```
7.490910297239916
```

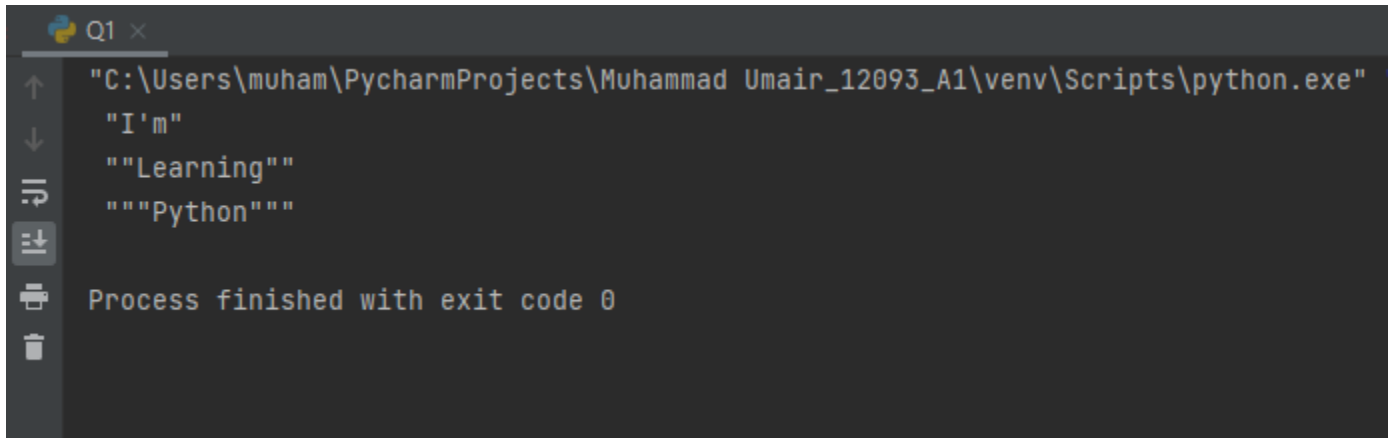
```
10.009131205673757
```

Question 1

Code:

```
print(" \"I'm\" \n \"\"Learning\"\" \n \"\"Python\"\" \"")
```

Output:



```
Q1 x
↑ "C:\Users\muham\PycharmProjects\Muhammad Umair_12093_A1\venv\Scripts\python.exe"
↓ "I'm"
↺ ""Learning""
↻ ""Python""
⌵
🖨 Process finished with exit code 0
🗑
```

Question 2

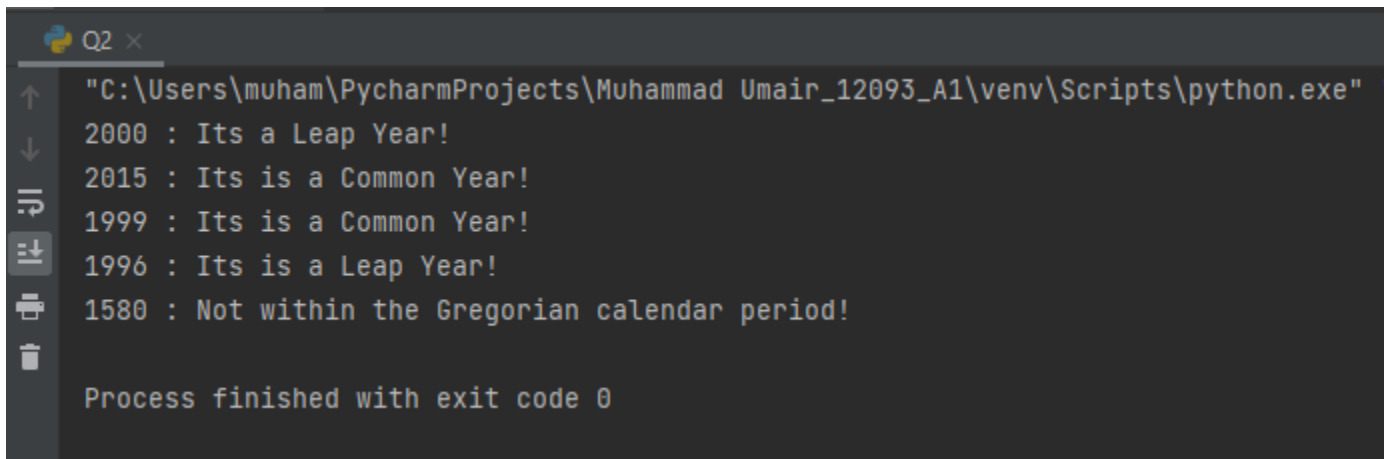
Code:

```
years_list = [2000, 2015, 1999, 1996, 1580]

def check_calender(year):
    if year > 1582:
        if year % 4 != 0:
            return "Its is a Common Year!"
        elif year % 100 != 0:
            return "Its is a Leap Year!"
        elif year % 400 != 0:
            return "Its is a Common Year!"
        else:
            return "Its a Leap Year!"
    else:
        return "Not within the Gregorian calendar period!"

for i in years_list:
    print(i, ":", check_calender(i))
```

Output:



```
Q2 x
"C:\Users\muham\PycharmProjects\Muhammad Umair_12093_A1\venv\Scripts\python.exe"
2000 : Its a Leap Year!
2015 : Its is a Common Year!
1999 : Its is a Common Year!
1996 : Its is a Leap Year!
1580 : Not within the Gregorian calendar period!

Process finished with exit code 0
```

Question 3

Code:

```
metres_per_miles = 1609.344
litres_per_gallon = 3.785411784

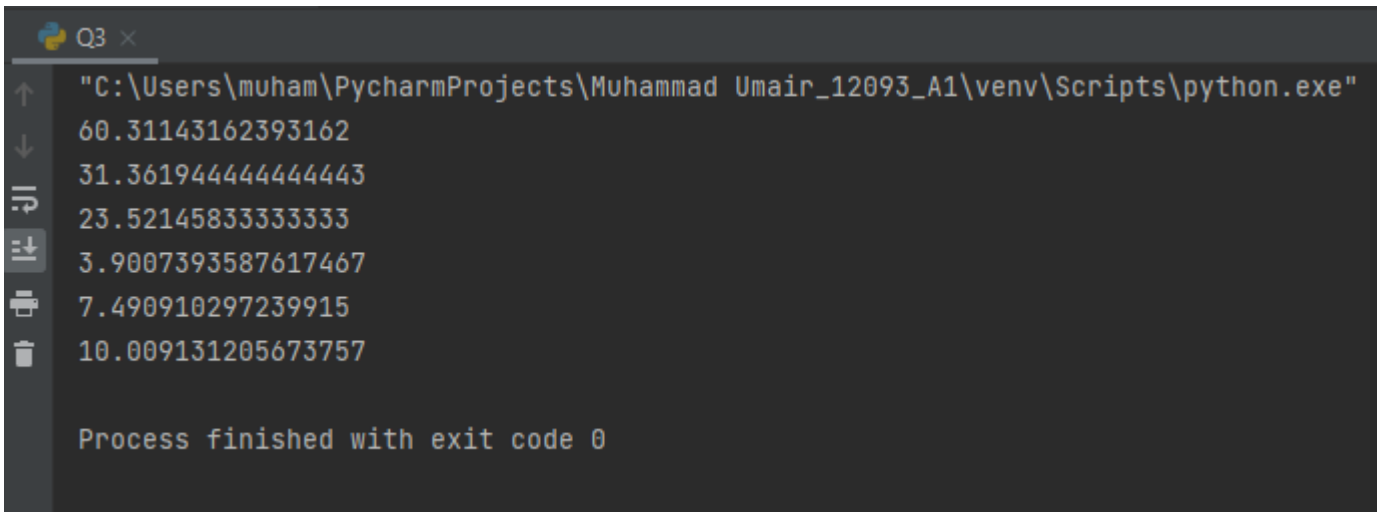
def l100kmtompg(litre) -> float:
    mpg = (100 * 1000 * litres_per_gallon) / (litre * metres_per_miles)
    return mpg

def mpgtol100km(mile) -> float:
    l100km = (100 * 1000 * litres_per_gallon) / (mile * metres_per_miles)
    return l100km

print(l100kmtompg(3.9))
print(l100kmtompg(7.5))
print(l100kmtompg(10))

print(mpgtol100km(60.3))
print(mpgtol100km(31.4))
print(mpgtol100km(23.5))
```

Output:



```
Q3 x
"C:\Users\muham\PycharmProjects\Muhammad Umair_12093_A1\venv\Scripts\python.exe"
60.31143162393162
31.361944444444443
23.521458333333333
3.9007393587617467
7.490910297239915
10.009131205673757

Process finished with exit code 0
```