HDS 802 – Python Homework 2

This homework covers working with conditionals and loops in Python. You are allowed to work together but are expected to write your own code. You are also encouraged to include comments in your code (comments can help in assigning partial credit if the function does not work properly). Please submit either a Python file (.py) or a Jupyter Notebook file (.jpynb).

1. Remove duplicates

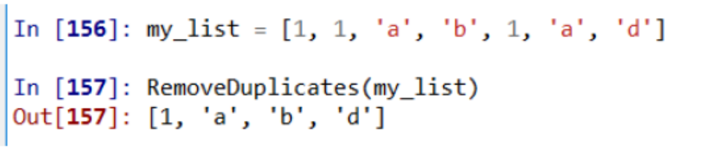
Write a Python function to remove duplicates from a list. Your function should not use any built-in Python function (it should use a for loop).

Function name: *RemoveDuplicates*

Parameters:
• A list

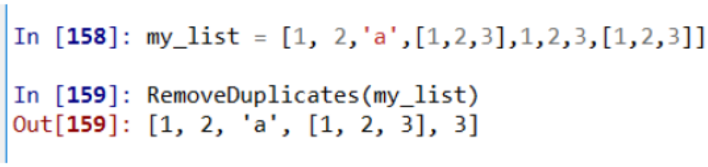
Return:
• A list with any duplicate items removed

Test cases:



```
In [156]: my_list = [1, 1, 'a', 'b', 1, 'a', 'd']

In [157]: RemoveDuplicates(my_list)
Out[157]: [1, 'a', 'b', 'd']
```



```
In [158]: my_list = [1, 2,'a',[1,2,3],1,2,3,[1,2,3]]

In [159]: RemoveDuplicates(my_list)
Out[159]: [1, 2, 'a', [1, 2, 3], 3]
```

2. Quadratic formula

The quadratic formula is used for solving quadratic equations of the form $ax2+bx+c$=0. Write a function named *Quadratic* which solves for the value(s) of x. There are 3 cases here: there may be 2 real solutions, 1 repeated real solution, or no real solutions. Your function should handle each differently.
The quadratic formula is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Function name: *Quadratic*

Parameters:

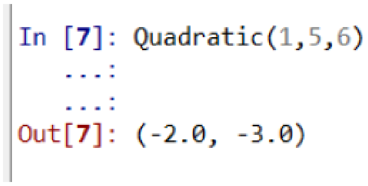
- a – the coefficient of the x^2 term in the equation
- b – the coefficient of the x term in the equation
- c – the constant term in the equation

Return:

- Either of the 2 real solutions to the equation
- If the solution is a repeated solution, it should only return one value
- If there are no real solutions, it should return a sentence saying that there are no real solutions (there is a very easy way to check if there are no real solutions)

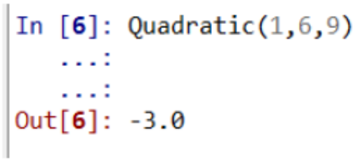
Three sample test cases are shown below:

$$x^2 + 5x + 6 = 0$$



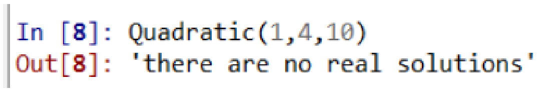
```
In [7]: Quadratic(1,5,6)
   ...:
   ...:
Out[7]: (-2.0, -3.0)
```

$$x^2 + 6x + 9 = 0$$



```
In [6]: Quadratic(1,6,9)
   ...:
   ...:
Out[6]: -3.0
```

$$x^2 + 4x + 10 = 0$$

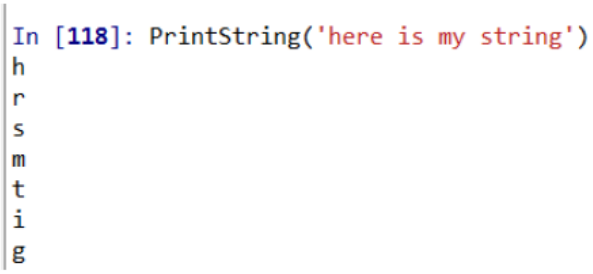


```
In [8]: Quadratic(1,4,10)
Out[8]: 'there are no real solutions'
```

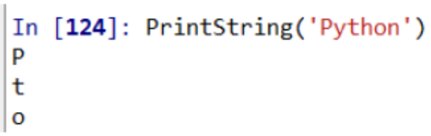
3. PrintString

Write a function named *PrintString* which prints every other letter of a string, beginning with the first letter. The function should accept one parameter, a string. Any spaces in the string should not be printed (but do not remove the white spaces from the string, see test cases below). Your function should have logic in it so that it does not print white spaces when it encounters any.

Three test cases are shown below:

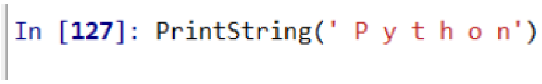


```
In [118]: PrintString('here is my string')
h
r
s
m
t
i
g
```



```
In [124]: PrintString('Python')
P
t
o
```

(nothing is printed for this 3rd test case)



```
In [127]: PrintString(' P y t h o n')
```

4. Fibonacci numbers

The Fibonacci numbers are a sequence of numbers where each number in the sequence is the sum of the previous 2 numbers. The first two numbers in the sequence are 1:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …

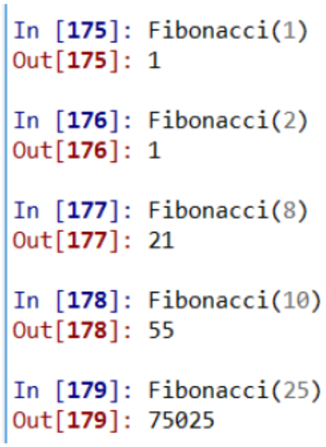
Your function must use a `for` loop for this.

Function name: *Fibonacci*

Parameters:
- n – a positive integer, representing the number in the sequence that is desired (the first number in the sequence is n=1)

Return:
- the corresponding Fibonacci number in the sequence

Test cases:



```
In [175]: Fibonacci(1)
Out[175]: 1

In [176]: Fibonacci(2)
Out[176]: 1

In [177]: Fibonacci(8)
Out[177]: 21

In [178]: Fibonacci(10)
Out[178]: 55

In [179]: Fibonacci(25)
Out[179]: 75025
```

5. Collatz conjecture

The Collatz conjecture is as follows:

> Take any positive integer n. If n is even, divide it by 2. If n is odd, multiply it by 3 and add 1. Repeat this process until you eventually get to the number 1. The conjecture states that this works for any positive integer n.

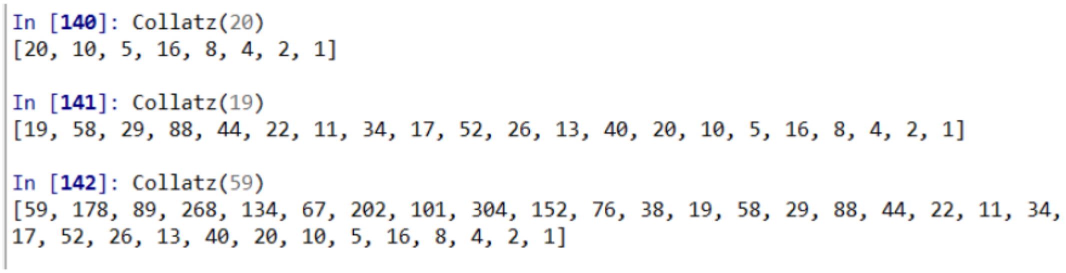
Write a function that takes any positive integer and performs these operations until the value 1 is reached.

Function name: *Collatz*

Parameters:
• n – a positive integer

Return:
• a list with all the numbers (beginning with the first one) until 1 is reached

Some sample test cases:



```
In [140]: Collatz(20)
[20, 10, 5, 16, 8, 4, 2, 1]

In [141]: Collatz(19)
[19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1]

In [142]: Collatz(59)
[59, 178, 89, 268, 134, 67, 202, 101, 304, 152, 76, 38, 19, 58, 29, 88, 44, 22, 11, 34,
17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1]
```