Tutorial 01
Python Overview

1. Complete the following Python function, `minmax(data)` such that it takes a sequence of one or more numbers, and returns the smallest and largest numbers, in the form of a tuple of length two.

   (**NOTE**: Do not use the built-in `min` or `max` in implementing your solution.)

```
def minmax(data):
    small = big = data[0]    # assuming data is non-empty

    for _____________________________________
        if _____________________________________
            _____________________________________
        elif _____________________________________
            _____________________________________

    return _____________________________________
```

2. Write a short Python function `sum_of_squares(n)` that takes a positive integer `n` and returns the sum of the squares of all the positive integers smaller than or equal to `n`.

3. Modify the `sum_of_squares(n)` function written for Qn. 2 such that it now takes a positive integer `n` and returns the sum of the squares of all the odd positive integers smaller than or equal to `n`.

   (**NOTE**: Do not use the modulus operator in implementing your solution.)

4. What parameters should be sent to the range constructor, to produce a range with values:
   a) 50, 60, 70, 80
   b) 8, 6, 4, 2, 0, -2, -4, -6, -8

5. Write a short Python function `num_vowels(text)` that counts the number of vowels in a given character string.

6. Write a Python program that repeatedly reads lines from standard input until an `EOFError` is raised, and then outputs those lines in reverse order.

   (**NOTE**: A user can indicate end of input by typing ctrl-D).
   (**HINTS**: You will need to use the `try` and `except` blocks to capture the exception. Use a list to store all the lines then try using the built-in `reverse()` function to reverse the order.)

Sample Output:

```
Enter a line (ctrl-D to stop): This is line 1
Enter a line (ctrl-D to stop): This is line 2
Enter a line (ctrl-D to stop): This is line 3
Enter a line (ctrl-D to stop): ^D
This is line 3
This is line 2
This is line 1

Process finished with exit code 0
```

7. Write a Python function that takes a sequence of numbers and determines if all the numbers are different from each other, i.e. they are distinct.

   (**HINT**: The simple solution just checks each number against every other one, but there are more efficient solutions around. For now, just implement the simple solution but make sure you don't compare a number to itself.)

-- End of Tutorial --