For Loops

Summary:

- A for loop is used for iterating over a sequence: it will execute a set of statements, once for each item in a list, tuple, string, etc.

  Example:

```
Letters = [a, b, c, d]

for x in Letters:

     print (x)
```

- Break statement: Stop the loop before it has looped through all the items.
- Continue statement: Stop the current iteration of the loop, and continue with the next.
- Range() function: Loop through a set of code a specified number of times. Syntax: range(start value, end value, steps). Default start value = 0, steps = 1.
- Else keyword in a **for** loop: Specifies the block of code which be executed after the loop is finished.
- A nested loop is a loop inside a loop. For each iteration of the outer loop, an inner loop will be executed once.
- Pass statement: To write a for loop with no content, without getting an error.

Exercises

Do all of the following using atleast one **for** loop. You might also need to use if-else statements for some questions.

Exercise 1:

Display numbers from -20 to -10.

Exercise 2:

Input a number from the user. Calculate the sum of all numbers from 1 to that given number.

Exercise 3:

```
list = [10, 20, 30, 40, 50, 60, 100, 150]
```

Print the following list in the revered order.

Exercise 4:

Print the following number pattern, using nested for loops:

```
1

1 2

1 2 3

1 2 3 4

1 2 3 4 5
```

Exercise 5:

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```

Write a program to display only those numbers from the given list that will satisfy the following conditions:

- The number must be divisible by five
- If the number is greater than 165, then skip it and move to the next number
- If the number is greater than 500, then stop the loop

Exercise 6:

Find the factorial of any number entered by the user.

***