Python - functions - Recursion

Factorial Calculation:

Implement a recursive function to compute the factorial of a non-negative integer

In [22]:

```
def fact(n):
    if n<=1:
        return 1
    else:
        return n*fact(n-1)
fact(6)
```

Out[22]:

```
720
```

Fibonacci Sequence:

Create a recursive function that returns the nth number in the Fibonacci sequence.

In [23]:

```
def fibonaci(n):
    if n<=2:
        return 1
    else:
        return fibonaci(n-1) + fibonaci(n-2)
fibonaci(5)
```

Out[23]:

```
5
```

Sum of Digits:

Write a recursive function to find the sum of the digits of a given positive integer

In [24]:

```
def sum_digits(n):
    if n==0:
        return 0
    else:
        return n%10 + sum_digits(n//10)
sum_digits(123)
```

Out[24]:

```
6
```

String Reversal:

Create a recursive function that takes a string as input and returns the string in reverse order.

In [25]:

```python
def str_rev(txt):
    if len(txt)==0:
        return ""
    else:
        return txt[-1] + str_rev(txt[:-1])
str_rev("LIRIL")
```

Out[25]:

```
'LIRIL'
```

Count Vowels:

Implement a recursive function to count the number of vowels in a given string.

In [26]:

```python
def count_vowels(txt):
    if len(txt) == 0:
        return 0
    else:
        return (1 if txt[-1].lower() in 'aeiou' else 0) + count_vowels(txt[:-1])
count_vowels("Alex")
```

Out[26]:

```
2
```

Flatten a List:

Create a recursive function to flatten a nested list structure into a single-level list.

In [27]:

```python
def flatten_list(l):
    if len(l) == 0:
        return []
    if isinstance(l[0], list):
        return flatten_list(l[0]) + flatten_list(l[1:])
    else:
        return [l[0]] + flatten_list(l[1:])

flatten_list([[1,2,3],[2,3,4, [5,6,7]]])
```

Out[27]:

```
[1, 2, 3, 2, 3, 4, 5, 6, 7]
```