In [1]:
```
# To calculate sum of N terms of an AP series
def sum_AP(start,difference,N):
    sum = 0
    error_msg = 'No. of terms must be an positive integer'
    if N > 0 and isinstance(N,int): # Checking whether no. of terms is a positive in
        for i in range(N):
            sum = start + i*difference
        return sum
    else:
        return error_msg
```

In [2]:
```
sum_AP(10,1.5,20)
```

Out[2]:
```
38.5
```

In [3]:
```
sum_AP(10,1.5,20.3)
```

Out[3]:
```
'No. of terms must be an positive integer'
```

In [4]:
```
# To calculate sum of N terms of an GP series
def sum_GP(start,ratio,N):
    sum = 0
    error_msg = 'No. of terms must be an positive integer'
    if N > 0 and isinstance(N,int): # Checking whether no. of terms is a positive in
        for i in range(N):
            sum = start*ratio**i
        return sum
    else:
        return error_msg
```

In [5]:
```
sum_GP(12,0.7,13)
```

Out[5]:
```
0.1660954464119999
```

In [6]:
```
sum_GP(1,-14,12.7)
```

Out[6]:
```
'No. of terms must be an positive integer'
```

In [7]:
```
# To calculate sum of N terms of an HP series
def sum_HP(start,difference,N):
    error_msg = 'Cannot start with 0'
    error_msg_2 = 'No. of terms must be an positive integer'
    if start == 0: # Start of an HP cannot be 0 since 1/0 is undefined
        return error_msg
    else:
        if N > 0 and isinstance(N,int): # Checking whether no. of terms is a positiv
            for i in range(N):
                sum = start + i*difference
            return sum
        else:
            return error_msg_2
```

In [8]:
```
sum_HP(4,2,54)
```

Out[8]:
```
110
```

In [9]:
```
sum_HP(1,2,4.5)
```

Out[9]:
```
'No. of terms must be an positive integer'
```

In [10]:
```
sum_HP(0,2,4)
```

Out[10]:
```
'Cannot start with 0'
```