ASSIGNMENT 2

In [4]:

```python
# WAP that finds greatest of 3 numbers using functions. Pass the numbers a

def find_num(a, b, c):
    if a>=b and a>=c:
        return a
    elif b>=a and b>=c:
        return b
    elif c>=a and c >=b:
        return c
    else:
        return "Error"

x = int(input("Enter 1st number: "))
y = int(input("Enter 2nd number: "))
z = int(input("Enter 3rd number: "))
r = find_num(x, y, z)
print(f'Greatest nyumber out of {x}, {y}, {z} is: {r}')
```

```
Enter 1st number: 87634
Enter 2nd number: 897
Enter 3rd number: 1
Greatest nyumber out of 87634, 897, 1 is: 87634
```

In [6]:

```python
# WAP to implement these formulae of permutations and combinations.
# Number of permutations of n objects taken r at a time: p(n, r) = n!/(n-r
# Number of combinations of n objects taken r at a time is: c(n, r) = n!/(


def fact(n):
    f = 1
    for i in range(2, n+1):
        f = f*i
    return f

n = int(input("Enter total number of items: "))
r = int(input("Enter number of selected items: "))

p = fact(n)/fact(n-r)
c = p / fact(r)

print("Number of permutations: ", p)
print("Number of combinations: ", c)
```

```
Enter total number of items: 5
Enter number of selected items: 2
Number of permutations:  20.0
Number of combinations:  10.0
```

In [9]:

```
# Write a function cubesum() that accepts an integer and returns the sum o
# Use this function to make functions PrintArmstrong() and isArmstrong() t

# Armstrong number 153 = 1^3+5^3+3^3

def cubeSum(n):
    s = 0
    a2 = n

    # Find sum
    while(a2 >= 1):
        s += (a2%10)**3
        a2 = int(a2/10)
    return s

def isArmstrong(n):
    if n == cubeSum(n):
        return True
    else:
        return False

def printArmstrong(n):
    if isArmstrong(n):
        print("Numer is Armstrong")
        print(n)
    else:
        print("Number is not Armstrong")

n = int(input("Enter the number: "))
printArmstrong(n)


```

```
Enter the number: 134
Number is not Armstrong
```

In [10]:

```
# WAP to create and print a list where the values are the squares of numbe

ls = []
for i in range(1, 31):
    ls.append(i * i)
print(ls)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 32
4, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900]
```

In [11]:

```python
# Given a string s = "1234" and an integer n = 5678. Concatenate them as a
# back to an integer. What is the final value?

s = "1234"
n = 5678

a = s + str(n)
print(a, type(a))

b = int(a)
print(b, type(b))
```

```
12345678 <class 'str'>
12345678 <class 'int'>
```

In [21]:

```python
# WAP that repeatedly asks the user to enter a positive integer. If the us
# should ask again until a positive integer is entered.

c = 0
while True:
    n = input("Enter a positive integer: ")
    if c >= 1 and int(n) > 0:
        print("Positive number entered again! Stopping!")
        break
    elif int(n) <= 0 and c < 1:
        print("Try Again!")
        c +=1

```

```
Enter a positive integer: 2
Enter a positive integer: 3
Enter a positive integer: 4
Enter a positive integer: 5
Enter a positive integer: 6
Enter a positive integer: 7
Enter a positive integer: 8
Enter a positive integer: 9
Enter a positive integer: 12
Enter a positive integer: 32
Enter a positive integer: -1
Try Again!
Enter a positive integer: 0
Enter a positive integer: 0
Enter a positive integer: 0
Enter a positive integer: -1
Enter a positive integer: -1
Enter a positive integer: 234
Positive number entered again! Stopping!
```