In [1]:
```
def addNumber(a,b):
    return a+b
sum = addNumber(10,20)
```

In [2]:
```
print(sum)
```

```
30
```

In [3]:
```
# here a,b are parameters and 10,20 are arguments
```

In [4]:
```
def returningMultipleValues():
    return 10,20,30

output = returningMultipleValues()
print(output)
```

```
(10, 20, 30)
```

In [5]:
```
def addNumbers():
    a = 10
    b = 20
    return (a+b)


sum = addNumbers()
```

In [6]:
```
print(sum)
```

```
30
```

In [8]:
```
def add():
    try:
        a = int(input())
    except ValueError as err:
        print(err)
    try:
        b = int(input())
    except ValueError as err:
        print(err)
    print(a+b)
add()
```

```
6
4
10
```

Fibonacci Series

In [15]:
```
def fibo(n):
    if(n<=0):
        print("Invalid Arguments")
        return -1
    if(n==1):
        print("0")
        return 1
    if(n==2):
        print(0," ",1)
        return -1
    f1 = 0
    f2 = 1
    f3 = f1+f2
    print(f1)
    print(f2)
    count = 3
    while(count<=n):
        print(f3)
        f1 = f2
        f2 = f3
        f3 = f1+f2
        count += 1

n = int(input("Enter n value: "))
fibo(n)
```

```
Enter n value: 10
0
1
1
2
3
5
8
13
21
34
```

LCM and HCF

In [27]:
```
a = int(input("Enter number1: "))
b = int(input("Enter number2: "))
def lcm(a,b):
    val = a if a>b else b
    while True:
        if val%a==0 and val%b==0:
            lcm = val;
            break
        val += 1
    return val

def hcf(a,b):
    hcf = 1
    val = a if a<b else b
    for i in range(1,val+1):
        if(a%i==0 and b%i==0):
            hcf = i
    print(hcf)

hcf(a,b)
lcm(a,b)
```

```
Enter number1: 4
Enter number2: 8
4
```

Out[27]:
```
8
```

In [24]:
```
def doJob(n):
    if(n<=2):
        return
    print(n)
    doJob(n-1)
    print(n)
doJob(10)
```

```
10
9
8
7
6
5
4
3
3
4
5
6
7
8
9
10
```

In [25]:
```
def job(n):
    if(n<=5):
        return
    job(n-1)
    print(n)
    job(n-1)
    print(n)
job(8)
```

```
6
6
7
6
6
7
8
6
6
7
6
6
7
8
```

Towers of Hanoi

In [ ]:
```

```