dictionaries, set, recursion/problem statements, oop

In [14]:
```
def fibo(n):
    if(n<=0):
        return -1
    if(n==1):
        return 0
    elif(n==2):
        return 1
    else:
        return fibo(n)+fibo(n-1)
print(fibo(4))
```

```
3
```

Fibonacci series using recursion

sum of fibonacci series

In [5]:
```
def sumf(val):
    sumn = 1
    return sumn+val

def sumfibo(n):
    if(n<=0):
        return -1
    if(n==1):
        return 0
    elif(n==2):
        return 1
    else:
        val = sumfibo(n-1)+sumfibo(n-2)
        return sumf(val)
print(sumfibo(6))
```

```
12
```

In [7]:
```
def sumfibo(n):
    if(n<=0):
        return -1
    elif(n==1):
        return 0
    elif(n==2):
        return 1
    else:
        fst = 0
        scnd = 1
        res = fst+scnd
        for i in range(0,n-2):
            next = fst+scnd
            res = fst+scnd
            fst = scnd
            scnd = next
        print(res)
sumfibo(10)
```

```
34
```

map, set, dictionaries

In [8]:
```
# set is an unordered and unindexed that does not allow dupliates
```

In [11]:
```
arr = [1,2,3,4,1,2,5,6,7,3,4]
myset = set(arr)
print(myset)
```

```
{1, 2, 3, 4, 5, 6, 7}
```

In [10]:
```
myset.add(8)  #it will add an element to the set
```

In [12]:
```
print(myset)
```

```
{1, 2, 3, 4, 5, 6, 7}
```

In [13]:
```
myset.clear()  #it will clear the elements of the set
```

In [14]:
```
print(myset)
```

```
set()
```

In [15]:
```
myset.copy()  #this method will return a copy of a set
print(myset)
```

```
set()
```

In [16]:
```
# dictionaries
# => key-value, ordered, changeable, not-duplicates
```

In [17]:
```
mydict = {
    "name" : "anish",
    "age" : 24,
    "occupation" : "student"
}
print(mydict)
```

```
{'name': 'anish', 'age': 24, 'occupation': 'student'}
```

In [18]:
```
print(type(mydict))
```

```
<class 'dict'>
```

In [19]:
```
print(mydict("name"))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [19], in <cell line: 1>()
----> 1 print(mydict("name"))

TypeError: 'dict' object is not callable
```

In [20]:
```
print(mydict["name"])
```

```
anish
```

In [21]:
```
mydict["name"] = "Mahammed Anish"
print(mydict)
```

```
{'name': 'Mahammed Anish', 'age': 24, 'occupation': 'student'}
```

In [23]:
```
updated_name = {
    "name" : "man"
}
mydict.update(updated_name)
print(mydict)
```

```
{'name': 'man', 'age': 24, 'occupation': 'student'}
```

In [24]:
```
print(mydict["name"])
```

```
man
```

oop

In [25]:
```
# logic should revolve around the concept of classes and instance of these cl
```

In [26]:
```
class Car:
    engineType = "Strongest Engine"
    numberOfTyres = 4
    numberOfWindow = 6
    isFridgeAvailable = True

    def getNumberOfWindows(self):
        return self.numberOfWindow

    def getNumberOfTyres(self):
        return self.numberOfTyres

car = Car()
print(car.getNumberOfWindows())
print(car.getNumberOfTyres())
```

```
6
4
```

In [ ]: