In [11]:
```
# Ques A
def LinearSearch(data,x):
    i=1
    while i <= len(data) and x != data[i]:
        i += 1
    if i <= len(data):
        location=data.index(i)+1
    else:
        location=0
    print(location)
data=[1,2,3,4,5,6]
x=4
LinearSearch(data,x)
```

```
3
```

In [1]:
```
# Ques B
def BinarySearch(data,x):
    i=1
    j=len(data)+1
    while i<j:
        m=(i+j)//2
        if x>data[m]:
            i=m+1
        else:
            j=m
    if x==data[m]:
        location=data.index(i)+1
    else:
        location=0
    print(location)
data=[1,2,3,4,5,6]
x=4
BinarySearch(data,x)
```

```
3
```

In [17]:

```
# Ques C
class List:
    def __init__(self):
        self.lst=[]
    def InsertAtFirst(self,x):
        self.lst.insert(0,x)
    def InsertAtEnd(self,x):
        self.lst.append(x)
    def DeleteFromFirst(self):
        self.lst.remove(self.lst[0])
    def DeleteFromEnd(self):
        self.lst.pop()
    def LinearSearch(self,x):
        for i in self.lst:
            if x == i:
                return self.lst.index(i)+1
        return False
    def BinarySearch(self,x):
        i=1
        j=len(self.lst)+1
        while i<j:
            m=(i+j)//2
            if x>self.lst[m]:
                i=m+1
            else:
                j=m
        if x==self.lst[m]:
            location=self.lst.index(i)+1
        else:
            location=0
        return location
    def isSorted(self):
        #a=False
        for i in self.lst:
            if self.lst[i] > self.lst[i+1]:
                return False
            else:
                return True
    def Search(self,x):
        if self.isSorted==True:
            self.BinarySearch(x)
        else:
            self.LinearSearch(x)
a=List()
a.InsertAtFirst(3)
a.InsertAtEnd(4)
a.InsertAtEnd(5)
a.InsertAtEnd(6)
#a.DeleteFromFirst()
#a.DeleteFromEnd()
print(a.LinearSearch(6))
print(a.BinarySearch(5))
print(a.Search(4))
#print(a.lst)
```

```
4
0
[3, 4, 5, 6]
```