

In [8]:

```
1 lst = [3, 4, 5, 6, 7, 8, 9]
2 high = len(lst)-1
3 low = 0
4 val = int(input())
5
6 while low <= high:
7     mid = (high+low)//2
8     if lst[mid] == val:
9         print("Value found")
10        break
11    if lst[mid] < val:
12        low = mid + 1
13    high = mid - 1
14 else:
15     print("Value not found")
16
```

```
4
Value found
```

In [13]:

```
1 #Using Function
2
3 def binary_search(lst,val,l,r):
4     while l <= r:
5         mid = (l+r)//2
6         if lst[mid] == val:
7             return mid
8         elif lst[mid] < val:
9             l = mid + 1
10        else:
11            r = mid - 1
12    return -1
13
14 lst = [3, 4, 5, 6, 7, 8, 9]
15 high = len(lst)-1
16 low = 0
17 val = int(input())
18
19 x = binary_search(lst,val,low,high)
20
21 if x == -1:
22     print("Element not found")
23 else:
24     print("Element found")
```

```
3
Element found
```

In [20]:

```
1  #Recursion
2
3  def b_search(lst,x,low,high):
4
5      if high >= low:
6
7          mid = (high+low)//2
8
9          # If found at mid, then return it
10         if lst[mid] == x:
11             return mid
12
13         # Search the left half
14         elif lst[mid] > x:
15             return b_search(lst, x, low, mid-1)
16
17         # Search the right half
18         else:
19             return b_search(lst, x, mid + 1, high)
20
21     else:
22         return -1
23 lst = [3, 4, 5, 6, 7, 8, 9]
24 high = len(lst)-1
25 low = 0
26 val = int(input())
27 x = b_search(lst,val,low,high)
28
29 if x == -1:
30     print("Element not found")
31 else:
32     print("Element found")
```

```
3
Element found
```