Linear Search On Arrays

```
In [1]: def LinearSearch(arr,target):
            for idx,ele in enumerate(arr):
                if ele ==  target:
                    return idx
            return -1
```

```
In [2]: arr = [3,5,6,8,9]
        target = 8
        LinearSearch(arr,target)
        #ans 3

Out[2]: 3
```

```
In [3]: arr = [3,5,6,8,9]
        target = 10
        LinearSearch(arr,target)
        #ans -1

Out[3]: -1
```

Linear Search On Linked Lists

```
In [5]: #creation of linkedlist

        class node(object):
            def __init__(self,value):
                self.value = value
                self.nextnode = None

        a = node(1)
        b = node(4)
        c = node(3)
        d = node(6)

        a.nextnode = b
        b.nextnode = c
        c.nextnode = d
        d.nextnode = None
```

```
In [6]: def LinearSearch_LL(head,target):
            if head:
                while head:
                    if head.value == target:
                        return head
                    else:
                        head = head.nextnode
            return head
```

```
In [16]: target = 3
         LinearSearch_LL(a,target)
         # ans <__main__.node at 0x7f5298261eb8>
         LinearSearch_LL(a,target).value
         # ans = 3

Out[16]: 3
```

In [17]:
```
target = 9
LinearSearch_LL(a,target)
# ans = None cannot access value here
```

Complexity

worst case = O(n) average case = O(n) Best case = O(1)

In [ ]:
```
```