Linked List

```
# Node class design
class Node:
  def __init__(self, e, n):
    self.element = e
    self.next = n

```

```
10 20 30 40 50
```

```
# Creating a list
def createList(a):
  head = Node(a[0], None)
  tail = head
  for i in range(1, len(a)):
    n = Node(a[i], None)
    tail.next = n
    tail = tail.next
  return head
```

```
# Iteration over a linked list
def iteration(head):
  temp = head
  while temp != None:
    print(temp.element)
    temp = temp.next
```

```
# Counting number of element in the list
def count(head):
  count = 0
  temp = head
  while temp != None:
    count += 1
    temp = temp.next
  return count
```

```
# Getting element of an specific index
def elemAt(head, idx):
  count = 0
  temp = head
  obj = None
  while temp != None:
    if count == idx:
      obj = temp.element
      break
    temp = temp.next
  if obj == None:
    print("Invalid index")
  return obj
```

```
# Getting node of an specific index
def nodeAt(head, idx):
  count = 0
  temp = head
  obj = None
  while temp != None:
    if count == idx:
      obj = temp
      break
```

```
    temp = temp.next
  if obj == None:
    print("Invalid index")
  return obj


# Getting index of an specific element
def indexOf(head, elem):
  temp = head
  count = 0
  obj = None
  while temp != None:
    if temp.element == elem:
      obj = temp
      break
    count += 1
  if obj == None:
    print("Element not found")
  else:
    return count

```