```
# Iterative Python program to search
# an element in linked list

# Node class
class Node:

    # Function to initialise the
    # node object
    def __init__(self, data):

        # Assign data
        self.data = data

        # Initialize next as null
        self.next = None

# Linked List class
class LinkedList:
    def __init__(self):

        # Initialize head as None
        self.head = None

    # This function insert a new node at the
    # beginning of the linked list
    def push(self, new_data):

        # Create a new Node
        new_node = Node(new_data)

        # 3. Make next of new Node as head
        new_node.next = self.head

        # 4. Move the head to point to new Node
        self.head = new_node

    # This Function checks whether the value
    # x present in the linked list
    def search(self, x):

        # Initialize current to head
        current = self.head

        # Loop till current not equal to None
        while current != None:
            if current.data == x:

```

```
                # Data found
                return True

            current = current.next

        # Data Not found
        return False

# Driver code
if __name__ == '__main__':

    # Start with the empty list
    llist = LinkedList()

    # Use push() to construct list
    # 14->21->11->30->10
    llist.push(10);
    llist.push(4);
    llist.push(9);
    llist.push(35);
    llist.push(14);


    if llist.search(int(input())):
        print("Found")
    else:
        print("Not Found")

```

```
10
Found
```

Linear Search

```
def linearSearch(array, n, x):

    # Going through array sequencially
    for i in range(0, n):
        if (array[i] == x):
            return i
    return -1


array = [2, 4, 0, 1, 9]
x = int(input())
n = len(array)
result = linearSearch(array, n, x)
if(result == -1):
    print("Element not found")
```

```
        print("Element not found")
    else:
        print("Element found at index: ", result)
```

```
4
Element found at index:  1
```

INSERSION SORT

```
def insertionSort(array):
  for step in range(1, len(array)):
    key = array[step]
    j = step - 1
    while j >= 0 & key < array[j]:
      array[j + 1] = array[j]
      j = j - 1
    array[j + 1] = key
data = [9, 5, 1, 4, 3]
insertionSort(data)
print("InsertionSort", data)
```

```
InsertionSort [3, 4, 1, 5, 9]
```

Selection Sort

```
def selectionSort(array, size):
  for step in range(size):
       min_idx = step
       for i in range(step + 1, size):
           if array[i] < array[min_idx]:
               min_idx = i
       (array[step], array[min_idx]) = (array[min_idx], array[step])
data = [-2, 45, 0, 11, -9]
size = len(data)
selectionSort(data, size)
print(data)
```

```
[-9, -2, 0, 11, 45]
```

Merge Sort

```
def merge_sort(unsorted_array):
    if len(unsorted_array) > 1:
        mid = len(unsorted_array) // 2  # Finding the mid of the array
        left = unsorted_array[:mid]  # Dividing the array elements
        right = unsorted_array[mid:]  # into 2 halves
```

```
6
7           merge_sort(left)
8           merge_sort(right)
9
10          i = j = k = 0
11
12          #  data to temp arrays L[] and R[]
13          while i < len(left) and j < len(right):
14              if left[i] < right[j]:
15                  unsorted_array[k] = left[i]
16                  i += 1
17              else:
18                  unsorted_array[k] = right[j]
19                  j += 1
20              k += 1
21
22          # Checking if any element was left
23          while i < len(left):
24              unsorted_array[k] = left[i]
25              i += 1
26              k += 1
27
28          while j < len(right):
29              unsorted_array[k] = right[j]
30              j += 1
31              k += 1
32
33
34  def print_list(array1):
35      for i in range(len(array1)):
36          print(array1[i], end=" ")
37      print()
38
39  if __name__ == '__main__':
40      array = [20, 30, 60, 40, 10, 50]
41      print("Given array is", end="\n")
42      print_list(array)
43      merge_sort(array)
44      print("Sorted array is: ", end="\n")
45      print_list(array)
```

```
Given array is
20 30 60 40 10 50
Sorted array is:
10 20 30 40 50 60
```