```
Q7="7.1. Write a python program to implement insertion sort using
lists."
print(Q7)

def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and arr[j] > key:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

# Example usage:
nums = [8, 7, 13, 1, -9, 4]
insertion_sort(nums)
print("Insertion sort result:", nums)
```

```
7.1. Write a python program to implement insertion sort using lists.
Insertion sort result: [-9, 1, 4, 7, 8, 13]
```

```
Q7="7.2.  Write a python program to implement merge sort using lists."
print(Q7)

def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])
    return merge(left, right)

def merge(left, right):
    result = []
    i = j = 0
    while i < len(left) and j < len(right):
        if left[i] <= right[j]:
            result.append(left[i])
            i += 1
        else:
            result.append(right[j])
            j += 1
    result.extend(left[i:])
    result.extend(right[j:])
    return result

# Example usage
lst = [5, 2, 9, 1, 5, 6]
```

```
print("Sorted Merge sort list:", merge_sort(lst))
```

---

7. Write a python program to implement merge sort using lists.

```
Sorted Merge sort list: [1, 2, 5, 5, 6, 9]
```
