a) Develop a program to implement Merge sort.

```
def merge_sort(arr):
    if len(arr) <= 1:
        return arr

    mid = len(arr) // 2
    left_half = arr[:mid]
    right_half = arr[mid:]

    left_half = merge_sort(left_half)
    right_half = merge_sort(right_half)

    return merge(left_half, right_half)


def merge(left_half, right_half):
    result = []
    i = j = 0

    while i < len(left_half) and j < len(right_half):
        if left_half[i] < right_half[j]:
            result.append(left_half[i])
            i += 1
        else:
            result.append(right_half[j])
            j += 1

    result += left_half[i:]
    result += right_half[j:]

    return result


# Get user input for array

arr = input("Enter the array elements separated by spaces: ")
arr = [int(x) for x in arr.split()]
```

```
 Enter the array elements separated by spaces: 12 34 13 44 66 33 22 1 3 5
```

```
# Sort array using merge sort

sorted_arr = merge_sort(arr)



# Print sorted array

print("Sorted array:", sorted_arr)
```

✓ 0s completed at 11:53 PM

Sorted array: [1, 3, 5, 12, 13, 22, 33, 34, 44, 66]

b) Develop a program to implement Binary search.

```
def binary_search(arr, target):
    left = 0
    right = len(arr) - 1

    while left <= right:
        mid = (left + right) // 2

        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1


# Get user input for array
arr = input("Enter the sorted array elements separated by spaces: ")
arr = [int(x) for x in arr.split()]

# Get user input for target element
target = int(input("Enter the target element: "))

# Perform binary search on array
result = binary_search(arr, target)

# Print search result
if result == -1:
    print("Target element not found in array")
else:
    print(f"Target element found at index {result}")
```

```
Enter the sorted array elements separated by spaces: 3 5 7 9 33 55 77
Enter the target element: 33
Target element found at index 4
```