8-1-24

PAGE NO.:
DATE: / /

Leet Code Challenge.
WAP to find first and last position of element in sorted array.

```
#include <stdio.h>
#include <stdlib.h>

void bubbleSort(int nums[], int numsSize)
{
  for (int i=0; i< numsSize -1 ; i++)
  {
    for(int j=0; j<numsSize -i -1 ; j++)
    {
    //Swap if the element found is
    greater than the next element
    if (nums [j] > nums[j+1])
    {
      int temp = nums[j];
      nums [j] = nums[j+1];
      nums [j+1] = temp;
    }
  }
  }
}

int * searchRange(int nums[], int
    numsSize, int target, int* returnSize)
{
  int * result = (int *) malloc (2* sizeof(int));
  result [0] = -1;
  result [1] = -1;
  int left = 0;
```

```
int right = numsSize - 1;
while (left <= right)
{
int mid = left + (right - left) / 2;
if (nums[mid] == target)
 {
 // Found target, search for
  the range
 int start = mid;
 int end = mid;
 while (start > 0 && nums[start - 1]
                  == target)
  start--;
 while (end < numsSize - 1 &&
                  nums[end + 1] == target)
  end++;
 result[0] = start;
 result[1] = end;
 break;
 }
else if (nums[mid] < target)
  left = mid + 1;
else
  right = mid - 1;
}
*returnSize = 2;
return result;
}

int main()
{
 int numsSize;
```

```
// Get the size of the array
   from the user
printf("Enter the size of the array:");
scanf("%d", &numsSize);
int nums[numsSize];
// Get elements of the array from
   the user
printf("Enter %d elements of the
      array: \n", numsSize);
for (int i=0; i<numsSize; i++)
    scanf("%d", &nums[i]);

// Sort the array using bululeSort
bululeSort(nums, numsSize);

// Print the sorted array
printf("\nSorted Array: ");
for (int i=0; i<numsSize; i++)
    printf("%d", nums[i]);
printf("\n");

// Enter target element
int target;
printf("\nEnter target element: ");
scanf("%d", &target);
int returnSize;
int *result=searchRange(nums,
        numsSize, target, &returnSize);

printf("Result: [%d, %d] \n",
        result[0], result[1]);
free(result);
```

```
    return 0;
}
```

Output:

- Enter the size of the array: 5
  Enter 5 elements of the array:
  3 3 4 3 3
  Sorted Array : 3 3 3 3 4

  Enter target element: 3
  Result : [0, 3]

- Enter the size of the array: 1
  Enter 1 elements of the array:
  2
  Sorted Array : 2
  Enter target element: 2
  Result : [0, 0]

- Enter the size of the array: 2
  Enter 2 elements of the array:
  4 5
  Sorted Array : 4 5
  Enter target element: 6
  Result : [-1, -1]