DAY-4-DA

1) Write a C program to implement Linear Search Algorithm.

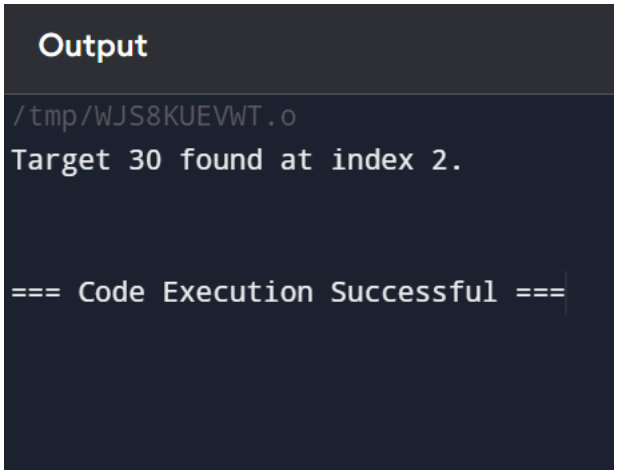
```
#include <stdio.h>

int linearSearch(int arr[], int size, int target) {
   for (int i = 0; i < size; i++) {
      if (arr[i] == target) {
         return i; // Return the index if the target is found
      }
   }
   return -1; // Return -1 if the target is not found
}

int main() {
   int arr[] = {10, 20, 30, 40, 50};
   int size = sizeof(arr) / sizeof(arr[0]);
   int target = 30;

   int result = linearSearch(arr, size, target);
   if (result == -1) {
      printf("Target %d not found in the array.\n", target);
   } else {
      printf("Target %d found at index %d.\n", target, result);
   }

   return 0;
}
```



2)Write a C program to implement Binary Search Algorithm.

```
#include <stdio.h>

// Function to perform binary search on a sorted array
int binarySearch(int arr[], int size, int target) {
   int left = 0;
   int right = size - 1;

   while (left <= right) {
      int mid = left + (right - left) / 2; // Calculate mid-point

      // Check if the target is present at mid
      if (arr[mid] == target) {
         return mid; // Target found
      }

      // If target is greater, ignore the left half
      if (arr[mid] < target) {
         left = mid + 1;
      } else {
```

```
            // If target is smaller, ignore the right half
            right = mid - 1;
        }
    }

    return -1; // Target not found
}

// Main function to test binary search
int main() {
    int arr[] = {2, 3, 4, 10, 40}; // Sorted array
    int size = sizeof(arr) / sizeof(arr[0]);
    int target;

    printf("Enter the number to search: ");
    scanf("%d", &target);

    int result = binarySearch(arr, size, target);

    if (result != -1) {
        printf("Element found at index: %d\n", result);
    } else {
        printf("Element not found in the array.\n");
    }

    return 0;
}
```

Output

```
/tmp/ecwVkEFFob.o
Enter the number to search: 3
Element found at index: 1


=== Code Execution Successful ===
```