**Institute of Computer Technology**

**B. Tech. Computer Science and Engineering**

**Sub: DS**

**Course Code: 2CSE302**

**Practical – 22**

**Name: Jaymin Gondaliya**

**Enrollment No: 23162171007**

**Sem - 3**

**Branch: CS**

**Class: A**

**Batch: 32**

**Imagine you’re looking for a card with the number &quot;42&quot; in two different situations.In the first case, the cards are randomly scattered in a bag. In the second case, the cards are neatly arranged in order from smallest to largest.**

**Definition: If the cards are unsorted, you’d use a Linear Search—picking each card one by one until you find &quot;42&quot; or have checked every card. This method is straightforward but can be slow since it may require examining every single card.**

**However, if the cards are sorted, you can use a Binary Search. Here, you’d start by dividing the ordered deck in half, checking the middle card. If it’s less than you’d ignore the lower half and search the upper half; if it’s more, you’d search the lower half. By repeatedly halving the search area, you quickly zero in on without needing to check each card individually.**

**Code:**

*#include* <stdio.h>

*//* **Function for Linear Search**

void linearSearch(int *arr*[], int *n*, int *target*) {

    printf("Performing Linear Search...\n");

*for* (int i = 0; i < *n*; i++) {

*if* (*arr*[i] == *target*) {

            printf("Found %d at index %d using Linear Search.\n", *target*, i);

*return*;

        }

    }

    printf("%d not found using Linear Search.\n", *target*);

}

*//* **Function for Binary Search**

void binarySearch(int *arr*[], int *n*, int *target*) {

    int left = 0, right = *n* - 1, mid;

    printf("Performing Binary Search...\n");

*while* (left <= right) {

        mid = left + (right - left) / 2;*//* **Calculate middle index**

*if* (*arr*[mid] == *target*) {

            printf("Found %d at index %d using Binary Search.\n", *target*, mid);

*return*;

        } *else* *if* (*arr*[mid] < *target*) {

            left = mid + 1;*//* **Search in the right half**

        } *else* {

            right = mid - 1;*//* **Search in the left half**

        }

    }

    printf("%d not found using Binary Search.\n", *target*);

}

int main() {

    int unsorted\_cards[] = {34, 21, 42, 5, 19};*//* **Unsorted cards for Linear Search**

    int sorted\_cards[] = {5, 19, 21, 34, 42};*//* **Sorted cards for Binary Search**

    int target = 42;

    int n\_unsorted = sizeof(unsorted\_cards) / sizeof(unsorted\_cards[0]);

    int n\_sorted = sizeof(sorted\_cards) / sizeof(sorted\_cards[0]);

*//* **Perform Linear Search on unsorted array**

    linearSearch(unsorted\_cards, n\_unsorted, target);

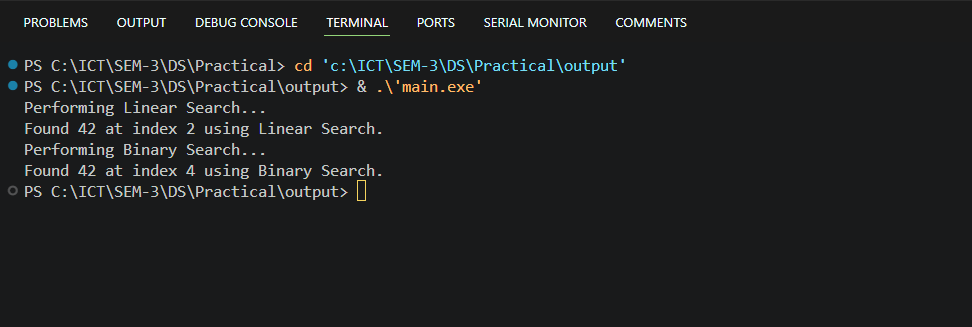
*//* **Perform Binary Search on sorted array**

    binarySearch(sorted\_cards, n\_sorted, target);

*return* 0;

}

**Output:**

****