

## **Experiment No.2**

### **Searching Techniques**

**You are building a contact manager app. A user wants to search for a contact either by scanning one by one or by using a fast lookup if the list is sorted.**

**Write a program that locates a specific name using both sequential and binary search techniques.**

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

#define SIZE 5

// Sequential Search
int sequentialSearch(char contacts[SIZE][20], char name[]) {
    int i;
    for(i = 0; i < SIZE; i++) {
        if(strcmp(contacts[i], name) == 0) {
            return i; // found
        }
    }
    return -1; // not found
}

// Binary Search (list must be sorted)
int binarySearch(char contacts[SIZE][20], char name[]) {
    int low = 0, high = SIZE - 1, mid;
    int result;

    while(low <= high) {
        mid = (low + high) / 2;
        result = strcmp(name, contacts[mid]);

        if(result == 0)
            return mid;
        else if(result < 0)
            high = mid - 1;
        else
            low = mid + 1;
    }
    return -1;
}

int main() {
    // Contact list (sorted alphabetically for binary search)
    char contacts[SIZE][20] = {"Amit", "Deepa", "Kiran", "Ravi", "Zoya"};
```

```
char name[20];
int choice, position;

printf("Enter the name to search: ");
scanf("%os", name);

printf("\nChoose search method:\n");
printf("1. Sequential Search\n");
printf("2. Binary Search (sorted list)\n");
printf("Enter your choice (1 or 2): ");
scanf("%d", &choice);

if(choice == 1) {
    position = sequentialSearch(contacts, name);
} else if(choice == 2) {
    position = binarySearch(contacts, name);
} else {
    printf("Invalid choice.\n");
    return 0;
}

if(position != -1) {
    printf("Contact '%s' found at position %d.\n", name, position + 1);
} else {
    printf("Contact '%s' not found.\n", name);
}

return 0;
}
```