



**Green University of Bangladesh**  
**Department of Computer Science and Engineering**  
**(CSE)**

**Faculty of Sciences and Engineering**  
**Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)**

**Lab Report NO: 02**  
**Course Title: Data Structure Lab**  
**Course Code: CSE 206**  
**Section: D8**

**Lab Experiment Name:** Basic operations of one-dimensional and two-dimensional array

**Student Details**

Name		ID
1.	Ashab Uddin	232002274

**Lab Date** : 11/09/24  
**Submission Date** : 20/09/24  
**Course Teacher's Name** : **Md. Parvez Hossain**

**Lab Report Status**

**Marks:** .....

**Signature:**

.....

**Comments:** ..... **Date:** .....

## 1. INTRODUCTION

The purpose of this lab report is to explore array searching techniques in C programming. We focus on how to efficiently find elements in an array using linear and binary search methods. Through these techniques, we aim to solve common real-world search problems and improve our understanding of algorithm efficiency.

## 2. OBJECTIVES

- Understand the concept of linear search.
- Explore the binary search algorithm.
- Learn how to implement the recursion technique.

## 3. IMPLEMENTATION

**Task 1:** Implement linear search algorithms using recursion.

**Solution:**

```
#include <stdio.h>

int linearSearch(int A[], int value, int i, int size) {
    if (i >= size) {
        return -1;
    }
    else if (A[i] == value) {
        return i;
    }
    return linearSearch(A, value, i + 1, size);
}

int main() {
    int A[] = {3,5,6,80,23,53,2,56,12,45};
    int value;
    int size ;
    size = sizeof(A) / sizeof(A[0]);

    printf("\nEnter the number to search: ");
    scanf(" %d", &value);

    int i=0;
```

```

    int result = linearSearch(A, value, i, size);

    if (result == -1) {
        printf("Value '%d' is not found in the ar.\n", value);
    } else {
        printf("Value '%d' is found at index %d.\n", value,
result);
    }
    return 0;
}

```

### Output:

```

PS D:\Fall 2024\DS Lab> cd "d:\Fall 2024\DS Lab\"

Enter the number to search: 45
Value '45' is found at index 9.

```

**Task 2:** Implement Linear Search for an array with character data using recursive method.

### Solution:

```

#include <stdio.h>

int linearSearchCharacter(char A[], char value, int i, int
size) {
    if (i >= size) {
        return -1;
    }
    else if (A[i] == value) {
        return i;
    }
    return linearSearchCharacter(A, value, i + 1, size);
}

int main() {
    char A[] = {'a', 's', 'h', 'b', 'd', 'u', 'i', 'n'};
    char value;
    int size = sizeof(A) / sizeof(A[0]);

```

```

printf("\nEnter the character to search: ");
scanf(" %c", &value);

int result = linearSearchCharacter(A, value, 0, size);

if (result == -1) {
    printf("Character '%c' is not found in the array.\n",
value);
} else {
    printf("Character '%c' is found at index %d.\n", value,
result);
}

return 0;
}

```

### Output:

```

Enter the character to search: u
Character 'u' is found at index 5.
PS D:\Fall 2024\DS Lab> cd "d:\Fall 2024\DS Lab\"

Enter the character to search: f
Character 'f' is not found in the array.
PS D:\Fall 2024\DS Lab> 
0

```

**Task 3:** Implement Binary Search for an array with character data using recursive method.

### Solution:

```

#include <stdio.h>

int binarySearchCharacter(char A[], char value, int L, int R) {

    if (L > R) {
        return -1;
    }
}

```

```

    int middle = L + (R - L) / 2;
    if (A[middle] == value) {
        return middle;
    }
    else if (A[middle] > value) {
        return binarySearchCharacter(A, value, L, middle - 1);
    }
    else{
        return binarySearchCharacter(A, value, middle + 1, R);
    }
}

int main() {

    char A[] = {'a', 'b', 'e', 'i', 'm', 'p', 'x', 'z'};
    char value;

    int size ;
    size = sizeof(A) / sizeof(A[0]);

    printf("\nEnter the character to search: ");
    scanf("%c", &value);
    int i=0;
    int result = binarySearchCharacter(A, value, i, size - 1);

    if (result == -1) {
        printf("Character '%c' is not found in the array.\n",
value);
    } else {
        printf("Character '%c' is found at index %d.\n", value,
result);
    }
    return 0;
}

```

**Output:**

```
Enter the character to search: p
Character 'p' is found at index 5.
PS D:\Fall 2024\DS Lab> cd "d:\Fall 2024\DS Lab\"

Enter the character to search: w
Character 'w' is not found in the array.
PS D:\Fall 2024\DS Lab> 
```

#### 4. DISCUSSION

In this lab report, we explored how to find elements in an array using recursion. We tackled three main problems. First, we used recursion with linear search to find a number in an array. Second, we applied the same approach to search for a character in a string. Lastly, we implemented binary search, which efficiently narrows down the search area by half with each step, making it ideal for sorted arrays or strings.