

Experiment No.1

Title : Binary Search and Applications

Reg.No.24141028

Program 1: Iterative Search

```
#include <stdio.h>

// Iterative Binary Search Function

int binarySearch(int arr[], int size, int target)

{
    int left = 0, right = size - 1;

    while (left <= right)

    {
        int mid = left + (right - left) / 2;

        if (arr[mid] == target)

            return mid;

        else if (arr[mid] < target)

            left = mid + 1;

        else

            right = mid - 1;
    }

    return -1; // Element not found
}
```

```
int main()
{
    int n, i, target;

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];

    printf("Enter %d sorted elements in ascending order:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

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

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

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

    return 0;
}
```

Output:

```
C:\Users\DELL\OneDrive\Desktop + | v
Enter the number of elements: 5
Enter 5 sorted elements in ascending order:
12 16 20 28 32
Enter the number to search: 28
Element found at index 3 (0-based indexing).

-----
Process exited after 16.05 seconds with return value 0
Press any key to continue . . . |
```

Program 2 : Recursive Search

```
#include <stdio.h>

// Recursive Binary Search with short name

int bSearch(int arr[], int l, int r, int x)

{
    if (l > r)
        return -1;
    int mid = l + (r - l) / 2;
    if (arr[mid] == x)
        return mid;
    else if (arr[mid] < x)
        return bSearch(arr, mid + 1, r, x);
    else
        return bSearch(arr, l, mid - 1, x);
}

int main()
{
```

```
int n, x;

printf("Enter number of elements: ");

scanf("%d", &n);

int arr[n];

printf("Enter %d sorted elements:\n", n);

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

    scanf("%d", &arr[i]);

printf("Enter element to search: ");

scanf("%d", &x);

int result = bSearch(arr, 0, n - 1, x);

if (result != -1)

    printf("Element found at index %d.\n", result);

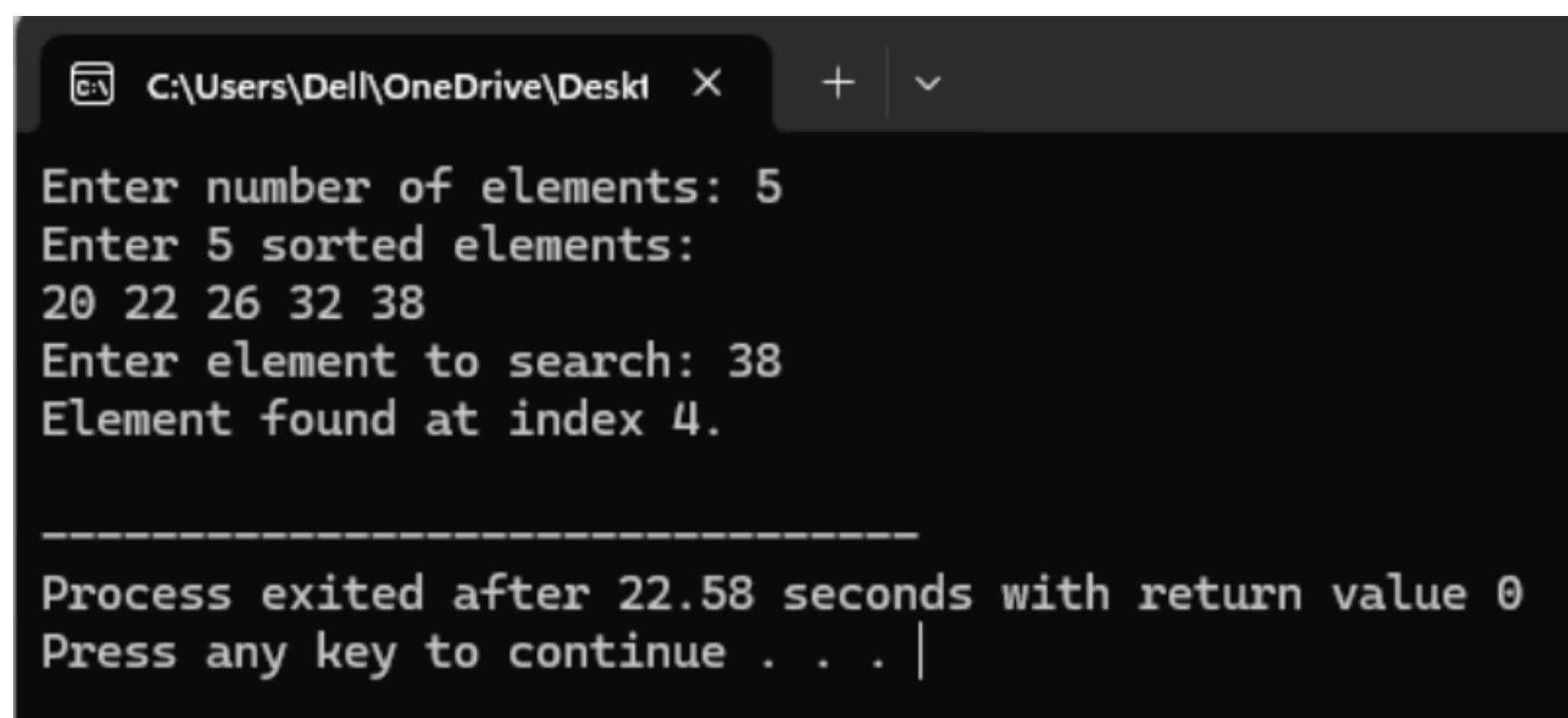
else

    printf("Element not found.\n");

return 0;

}
```

Output:



```
C:\Users\DELL\OneDrive\Desktop + | v

Enter number of elements: 5
Enter 5 sorted elements:
20 22 26 32 38
Enter element to search: 38
Element found at index 4.

-----
Process exited after 22.58 seconds with return value 0
Press any key to continue . . . |
```

Application Based on Binary Search :

Code : Library Management System

```
#include <stdio.h>
#include <string.h>
#define SIZE 5
int book_ids[SIZE] = {101, 203, 305, 408, 512};
char *book_titles[SIZE] =
{
    "C Programming",
    "Data Structures",
    "Operating Systems",
    "Computer Networks",
    "Database Systems"
};

// Binary search by book ID
int binary_search(int ids[], int size, int key)
{
    int low = 0, high = size - 1;
    while (low <= high)
    {
        int mid = (low + high) / 2;
        if (ids[mid] == key) return mid;
        else if (key < ids[mid]) high = mid - 1;
        else low = mid + 1;
    }
}
```

```
    }

    return -1; // Not Found
}

int main()
{
    int id;

    printf("Welcome to the Library System\n");
    printf("-----\n");

    while (1)
    {
        printf("\nEnter Book ID to search (-1 to exit): ");
        scanf("%d", &id);

        if (id == -1) break;

        int index = binary_search(book_ids, SIZE, id);
        if (index != -1) {

            printf("Book Found: [%d] %s\n", book_ids[index],
book_titles[index]);
        } else {

            printf("Book with ID %d not found.\n", id);
        }
    }

    printf("Exiting Library System.\n");

    return 0;
}
```

}

Output:

```
C:\Users\DELL\OneDrive\Desktop + | 
Welcome to the Library System
-----
Enter Book ID to search (-1 to exit): 305
Book Found: [305] Operating Systems

Enter Book ID to search (-1 to exit): 200
Book with ID 200 not found.

Enter Book ID to search (-1 to exit): -1
Exiting Library System.

-----
Process exited after 7.511 seconds with return value 0
Press any key to continue . . . |
```