

Linear search

```
#include <stdio.h>

// Function to perform a linear search
int linear_search(int a[], int n, int x)
{
    int i = 0;

    // Iterate through the array until the end or until the element is found
    while (i < n)
    {
        if (x == a[i])
            break; // Exit the loop if the element is found
        i++;
    }

    // If 'i' reaches 'n', the element was not found; return -1, otherwise,
    // return the position (+1) where it was found
    if (i == n)
        return -1;
    else
        return (i + 1);
}

int main()
{
    int n, x;

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

    // Declare an array of size 'n'
    int a[n];

    // Input the array elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    // Input the element to search for
    printf("Enter the element to search for: ");
    scanf("%d", &x);

    // Call the linear search function and store the result
```

```

int result = linear_search(a, n, x);

// Check if the element was found or not, and print the result
if (result == -1)
{
    printf("Element %d not found in the array.\n", x);
}
else
{
    printf("Element %d found at position %d.\n", x, result);
}

return 0;
}

```

Enter the number of elements in the array: 5

Enter 5 elements:

12

32

26

73

4

Enter the element to search for: 26

Element 26 found at position 3.

PS C:\Users\Mark Lopes\Desktop\college\ds> █

Linear search using recursion

```

#include <stdio.h>

// Function to perform a recursive linear search
int linear_search_rec(int a[], int n, int x)
{
    // Base case: If the array is empty (n == 0), the element is not found
    if (n == 0)
        return -1;
    // If the last element in the array matches the target element 'x', return
    // its position 'n'
    else if (a[n - 1] == x)
        return n;
    // If the last element does not match, recursively search the rest of the
    // array
    else
        return linear_search_rec(a, n - 1, x);
}

int main()
{

```

```

int n, x;

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

// Declare an array of size 'n'
int a[n];

// Input the array elements
printf("Enter %d elements:\n", n);
for (int i = 0; i < n; i++)
{
    scanf("%d", &a[i]);
}

// Input the element to search for
printf("Enter the element to search for: ");
scanf("%d", &x);

// Call the linear search function and store the result
int result = linear_search_rec(a, n, x);

// Check if the element was found or not, and print the result
if (result == -1)
{
    printf("Element %d not found in the array.\n", x);
}
else
{
    printf("Element %d found at position %d.\n", x, result);
}

return 0;
}

```

Enter the number of elements in the array: 5

Enter 5 elements:

12

32

42

5

90

Enter the element to search for: 5

Element 5 found at position 4.

PS C:\Users\Mark Lopes\Desktop\college\ds> █

Binary search

```
#include <stdio.h>

int binary_search(int a[], int low, int high, int ele)
{
    int mid;

    while (low <= high)
    {
        // Calculate the middle index
        mid = (low + high) / 2;

        // Check if the middle element is the target element
        if (ele == a[mid])
            return (mid + 1); // Element found, return its position

        // If the target element is less than the middle element,
        // search in the left half of the array
        else if (ele < a[mid])
            high = mid - 1;

        // If the target element is greater than the middle element,
        // search in the right half of the array
        else
            low = mid + 1;
    }

    // If the element is not found, return -1
    return -1;
}

int main()
{
    int n, ele;

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

    int a[n]; // Declare an array of size 'n'

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

    printf("Enter the element to search for: ");
```

```

scanf("%d", &ele);

int result = binary_search(a, 0, n - 1, ele);

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

return 0;
}

```

```

Enter the number of elements in the sorted array: 4
Enter 4 sorted elements:
12
23
32
46
Enter the element to search for: 23
Element 23 found at position 2.
PS C:\Users\Mark Lopes\Desktop\college\ds>

```

Binary search using recursion

```

#include <stdio.h>

// Function to perform a recursive linear search
int linear_search_rec(int a[], int n, int x)
{
    // Base case: If the array is empty (n == 0), the element is not found
    if (n == 0)
        return -1;
    // If the last element in the array matches the target element 'x', return
    // its position 'n'
    else if (a[n - 1] == x)
        return n;
    // If the last element does not match, recursively search the rest of the
    // array
    else
        return linear_search_rec(a, n - 1, x);
}

int main()

```

```

{
    int n, x;

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

    // Declare an array of size 'n'
    int a[n];

    // Input the array elements
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    // Input the element to search for
    printf("Enter the element to search for: ");
    scanf("%d", &x);

    // Call the linear search function and store the result
    int result = linear_search_rec(a, n, x);

    // Check if the element was found or not, and print the result
    if (result == -1)
    {
        printf("Element %d not found in the array.\n", x);
    }
    else
    {
        printf("Element %d found at position %d.\n", x, result);
    }

    return 0;
}

```

```

Enter the number of elements in the sorted array: 5
Enter 5 sorted elements:
12
34
53
70
97
Enter the element to search for: 97
Element 97 found at position 5.
PS C:\Users\Mark Lopes\Desktop\college\ds>

```