Assignment 9

Dated __________

Problem Statement

Write a C program to search for a particular element in an array using the **Linear Search** method. The program should allow the user to input an array of elements and then search for a specific element within it.

Input & Output Description

- **Input:**
    - The number of elements in the array (n), where n≤100n.
    - The elements of the array.
    - The element to be searched.
- **Output:**
    - If the element is found, display its position.
    - If the element is not found, display a message indicating that the element is not present in the array.

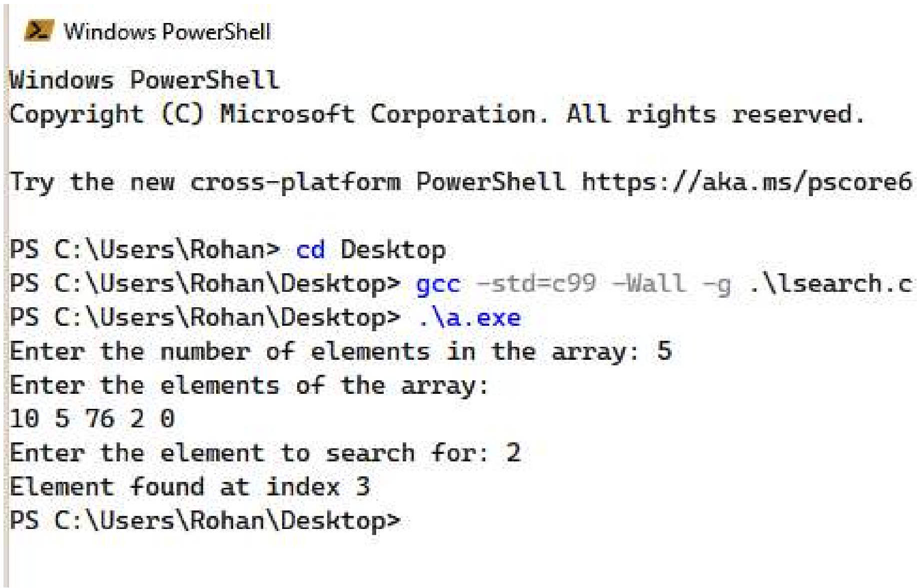
Algorithm for Linear Search

Step 1: Take input for the number of elements (n).
Step 2: Check if n exceeds the maximum allowed size (100). If true, display an error message and terminate the program.
Step 3: Take input for n elements and store them in an array (A[]).
Step 4: Take input for the search element (key).
Step 5: Initialize a loop counter i = 0.
Step 6: Repeat steps 7-8 until i < n.
Step 7: Check if A[i] == key. If true, print the position (i+1) and terminate the search.
Step 8: Increment i by 1 and continue searching.
Step 9: If the element is not found after the loop ends, display "Element not found" message.
Step 10: End.

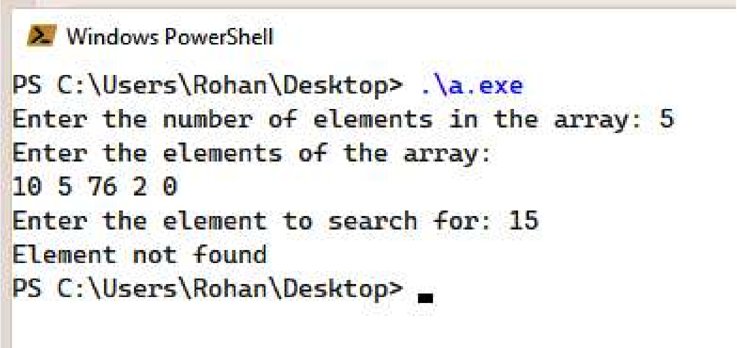
Source Code

```
#include <stdio.h>
#define MAX 100  // Defining the maximum size of the array

// Function for performing Linear Search
void linearSearch(int A[], int n, int key) {
    int i;
    for (i = 0; i < n; i++) {
        if (A[i] == key) {  // If key is found
            printf("\nElement %d found at position %d\n", key, i + 1);
            return;
        }
    }
    printf("\nElement %d not found in the array.\n", key);
}

// Main function
int main() {
    int A[MAX], n, key, i;

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

    // Checking if the size exceeds MAX
    if (n > MAX) {
        printf("\nSize exceeds the limit of %d. Cannot store elements.\n", MAX);
        return 1;
    }

    // Taking input for elements
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        printf("A[%d]: ", i);
        scanf("%d", &A[i]);
    }

    // Taking input for the element to search
    printf("\nEnter the element to search: ");
    scanf("%d", &key);

    // Calling Linear Search function
    linearSearch(A, n, key);

    return 0;
}
```

Output



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Rohan> cd Desktop
PS C:\Users\Rohan\Desktop> gcc -std=c99 -Wall -g .\lsearch.c
PS C:\Users\Rohan\Desktop> .\a.exe
Enter the number of elements in the array: 5
Enter the elements of the array:
10 5 76 2 0
Enter the element to search for: 2
Element found at index 3
PS C:\Users\Rohan\Desktop>
```



```
PS C:\Users\Rohan\Desktop> .\a.exe
Enter the number of elements in the array: 5
Enter the elements of the array:
10 5 76 2 0
Enter the element to search for: 15
Element not found
PS C:\Users\Rohan\Desktop> _
```

Discussion

It should be noted that the size of the stack for array is limited and is implementation-defined. So, there should be a limit.

Teacher's signature