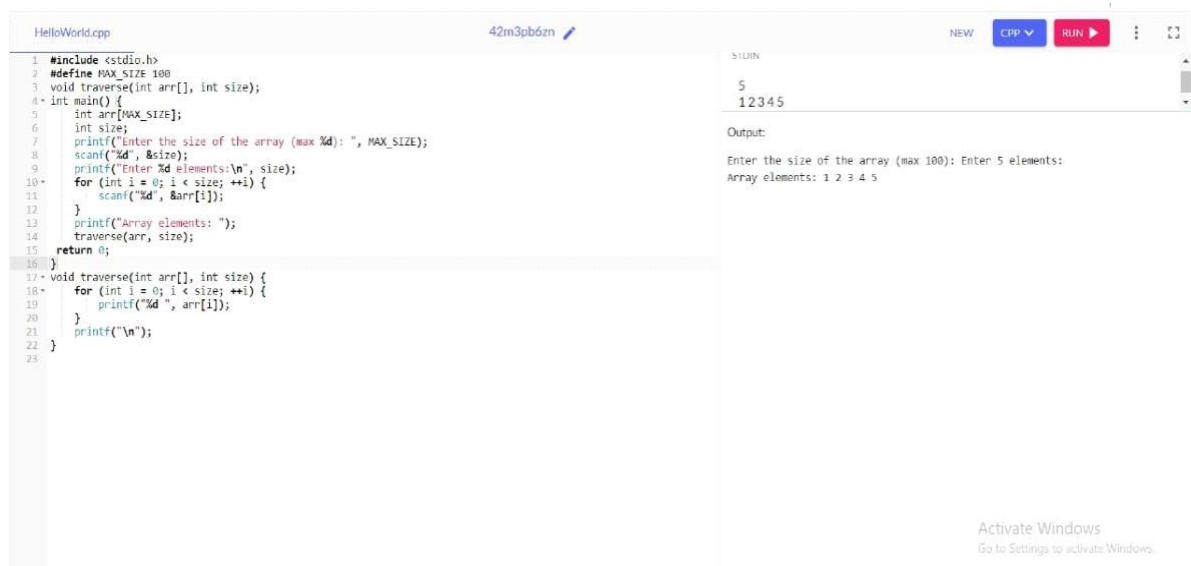


1. Write a C Program to implement following operations

a) traverse



The screenshot shows a C program in a code editor. The program defines a constant `MAX_SIZE` as 100 and a function `traverse` that iterates through an array and prints its elements. The `main` function prompts the user for the array size and elements, then calls `traverse`. The output shows the array elements 1, 2, 3, 4, and 5.

```
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 void traverse(int arr[], int size);
4 int main() {
5     int arr[MAX_SIZE];
6     int size;
7     printf("Enter the size of the array (max %d): ", MAX_SIZE);
8     scanf("%d", &size);
9     printf("Enter %d elements:\n", size);
10    for (int i = 0; i < size; ++i) {
11        scanf("%d", &arr[i]);
12    }
13    printf("Array elements: ");
14    traverse(arr, size);
15    return 0;
16 }
17 void traverse(int arr[], int size) {
18     for (int i = 0; i < size; ++i) {
19         printf("%d ", arr[i]);
20     }
21     printf("\n");
22 }
23
```

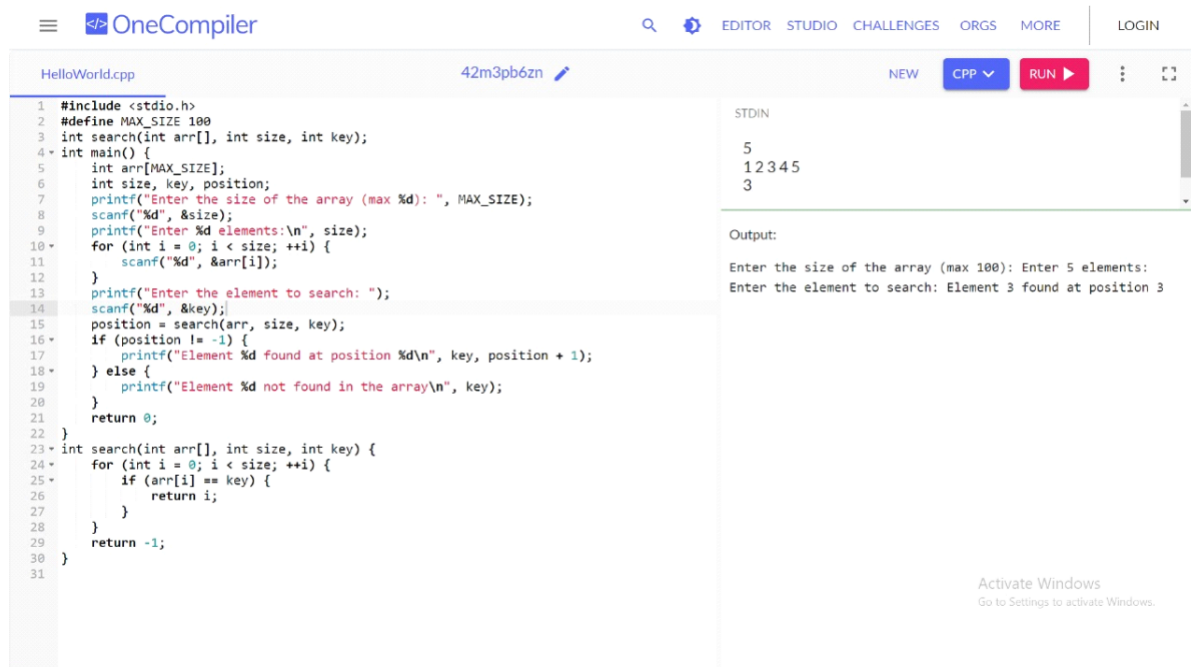
STDIN

```
5
12345
```

Output:

```
Enter the size of the array (max 100): Enter 5 elements:
Array elements: 1 2 3 4 5
```

b) search



The screenshot shows a C program in a code editor. The program defines a constant `MAX_SIZE` as 100 and a function `search` that iterates through an array to find a specific key. The `main` function prompts the user for the array size, elements, and a key to search for. The output shows the key 3 found at position 3.

```
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int search(int arr[], int size, int key);
4 int main() {
5     int arr[MAX_SIZE];
6     int size, key, position;
7     printf("Enter the size of the array (max %d): ", MAX_SIZE);
8     scanf("%d", &size);
9     printf("Enter %d elements:\n", size);
10    for (int i = 0; i < size; ++i) {
11        scanf("%d", &arr[i]);
12    }
13    printf("Enter the element to search: ");
14    scanf("%d", &key);
15    position = search(arr, size, key);
16    if (position != -1) {
17        printf("Element %d found at position %d\n", key, position + 1);
18    } else {
19        printf("Element %d not found in the array\n", key);
20    }
21    return 0;
22 }
23 int search(int arr[], int size, int key) {
24     for (int i = 0; i < size; ++i) {
25         if (arr[i] == key) {
26             return i;
27         }
28     }
29     return -1;
30 }
31
```

STDIN

```
5
12345
3
```

Output:

```
Enter the size of the array (max 100): Enter 5 elements:
Enter the element to search: Element 3 found at position 3
```

c) insert

HelloWorld.cpp
42m3pb6zn
NEW
CPP
RUN

```

1 #include <stdio.h>
2 #define MAX_SIZE 100
3 void insert(int arr[], int *size, int element, int position);
4 int main() {
5     int arr[MAX_SIZE];
6     int size, element, position;
7     printf("Enter the current size of the array (max %d): ", MAX_SIZE);
8     scanf("%d", &size);
9     printf("Enter %d elements:\n", size);
10    for (int i = 0; i < size; ++i) {
11        scanf("%d", &arr[i]);
12    }
13    printf("Enter the element to insert: ");
14    scanf("%d", &element);
15    printf("Enter the position to insert (1 to %d): ", size + 1);
16    scanf("%d", &position);
17    if (position < 1 || position > size + 1) {
18        printf("Invalid position to insert.\n");
19    } else {
20        insert(arr, &size, element, position - 1);
21        printf("Array after insertion: ");
22        for (int i = 0; i < size; ++i) {
23            printf("%d ", arr[i]);
24        }
25        printf("\n");
26    }
27    return 0;
28 }
29
30 void insert(int arr[], int *size, int element, int position) {
31     for (int i = *size - 1; i >= position; --i) {
32         arr[i + 1] = arr[i];
33     }
34     arr[position] = element;
35     *size += 1;
36 }
37

```

5
15694
2

Enter the array (max 100): Enter 5 elements:

ert: Enter the position to insert (1 to 6): Array after insertion: 1 5 2 6 9 4

Activate Windows
Go to Settings to activate Windows.

d) delete

HelloWorld.cpp
42m3pb6zn
NEW
CPP
RUN

```

1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int main() {
4     int array[MAX_SIZE];
5     int size, i, pos;
6     printf("Enter size of the array: ");
7     scanf("%d", &size);
8     printf("Enter elements of the array:\n");
9     for (i = 0; i < size; i++) {
10        scanf("%d", &array[i]);
11    }
12    printf("Enter the position of the element to delete (0-indexed): ");
13    scanf("%d", &pos);
14    if (pos < 0 || pos >= size) {
15        printf("Invalid position!\n");
16    } else {
17        for (i = pos; i < size - 1; i++) {
18            array[i] = array[i + 1];
19        }
20        size--;
21        printf("Array after deletion:\n");
22        for (i = 0; i < size; i++) {
23            printf("%d ", array[i]);
24        }
25        printf("\n");
26    }
27    return 0;
28 }
29
30

```

5
1020304050
2

Output:

Enter size of the array: Enter elements of the array:

Enter the position of the element to delete (0-indexed): Array

10 20 40 50

Activate Windows

e)update

The screenshot shows an online C++ compiler interface. The code editor on the left contains a C++ program named 'HelloWorld.cpp'. The program defines a constant 'MAX_SIZE' as 100, declares an array 'array' of size 'MAX_SIZE', and implements a 'main' function. The 'main' function prompts the user to enter the size of the array, then enters elements of the array. It then prompts for the position of the element to update (0-indexed). If the position is invalid (less than 0 or greater than or equal to the size), it prints an error message. Otherwise, it prompts for the new value, updates the array element at the specified position, and prints the array after updating. The output window on the right shows the program's execution: the user enters 5 for the size, then 3 5 7 8 9 for the elements, then 4 for the position to update, and finally 0 for the new value. The output shows the array after updating: 3 5 7 8 0.

```
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int main() {
4     int array[MAX_SIZE];
5     int size, i, pos, new_value;
6     printf("Enter size of the array: ");
7     scanf("%d", &size);
8     printf("Enter elements of the array:\n");
9     for (i = 0; i < size; i++) {
10         scanf("%d", &array[i]);
11     }
12     printf("Enter the position of the element to update (0-indexed): ");
13     scanf("%d", &pos);
14     if (pos < 0 || pos >= size) {
15         printf("Invalid position!\n");
16     } else {
17         printf("Enter the new value: ");
18         scanf("%d", &new_value);
19         array[pos] = new_value;
20         printf("Array after updating:\n");
21         for (i = 0; i < size; i++) {
22             printf("%d ", array[i]);
23         }
24         printf("\n");
25     }
26     return 0;
27 }
28 }
29 }
```

Output:

Enter size of the array: Enter elements of the array:
Enter the position of the element to update (0-indexed): Enter
3 5 7 8 0

2. Writing a recursive function to calculate the factorial of a number.

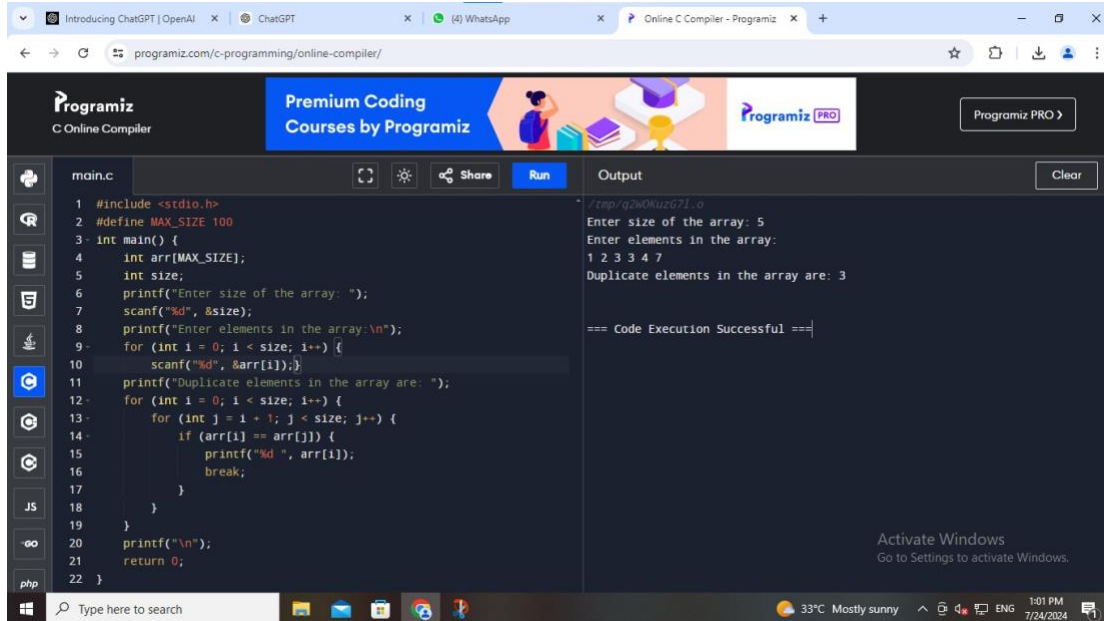
The screenshot shows an online C compiler interface. The code editor on the left contains a C program named 'main.c'. The program defines a recursive function 'factorial' that calculates the factorial of a non-negative integer 'n'. The 'main' function prompts the user to enter a non-negative integer, checks if it is negative, and if not, calls the 'factorial' function and prints the result. The output window on the right shows the program's execution: the user enters 5, and the output shows 'Factorial of 5 is 120'. The output also indicates 'Code Execution Successful'.

```
1 #include <stdio.h>
2 unsigned long long factorial(int n) {
3     if (n == 0) {
4         return 1;
5     } else {
6         return n * factorial(n - 1);
7     }
8 }
9 int main() {
10     int n;
11     printf("Enter a non-negative integer: ");
12     scanf("%d", &n);
13
14     if (n < 0) {
15         printf("Factorial is not defined for negative numbers.\n");
16     } else {
17         unsigned long long result = factorial(n);
18         printf("Factorial of %d is %llu\n", n, result);
19     }
20     return 0;
21 }
22 }
```

Output:

Enter a non-negative integer: 5
Factorial of 5 is 120
=== Code Execution Successful ===

3. Write a C Program to find duplicate element in an array



The screenshot shows the Programiz C Online Compiler interface. The code editor on the left contains a C program to find duplicate elements in an array. The output window on the right shows the program's execution results.

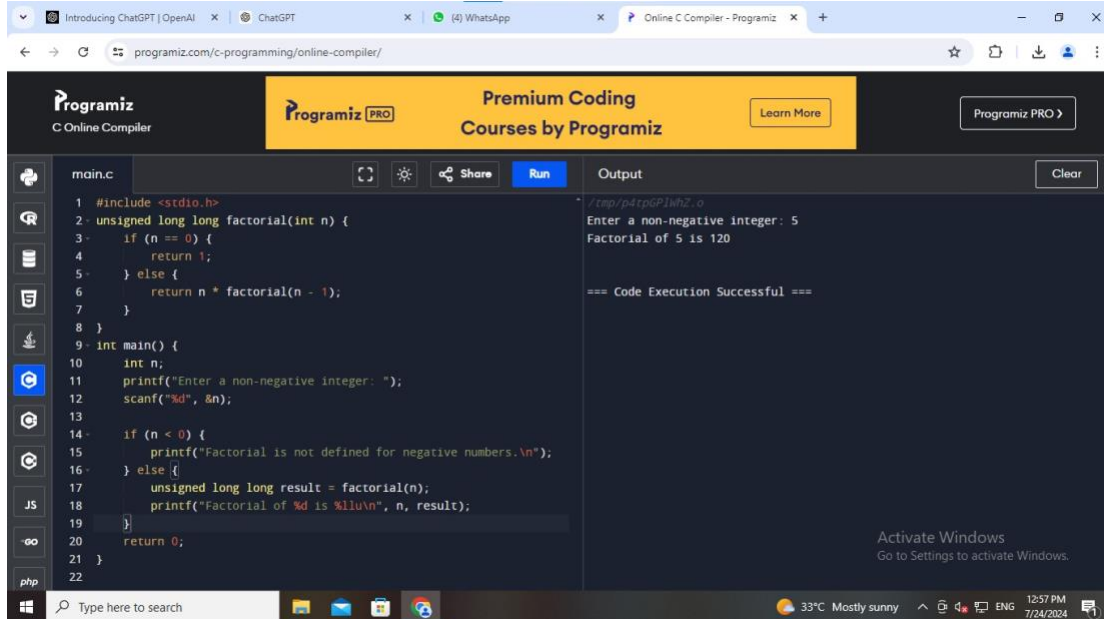
```
main.c
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int main() {
4     int arr[MAX_SIZE];
5     int size;
6     printf("Enter size of the array: ");
7     scanf("%d", &size);
8     printf("Enter elements in the array:\n");
9     for (int i = 0; i < size; i++) {
10         scanf("%d", &arr[i]);
11     }
12     printf("Duplicate elements in the array are: ");
13     for (int i = 0; i < size; i++) {
14         for (int j = i + 1; j < size; j++) {
15             if (arr[i] == arr[j]) {
16                 printf("%d ", arr[i]);
17                 break;
18             }
19         }
20     }
21     printf("\n");
22     return 0;
23 }
```

Output:

```
/tmp/q2w0kuzg71.o
Enter size of the array: 5
Enter elements in the array:
1 2 3 3 4 7
Duplicate elements in the array are: 3

=== Code Execution Successful ===
```

4. Write a C Program to find Max and Min from an array elements



The screenshot shows the Programiz C Online Compiler interface. The code editor on the left contains a C program to find Max and Min from an array elements. The output window on the right shows the program's execution results.

```
main.c
1 #include <stdio.h>
2 unsigned long long factorial(int n) {
3     if (n == 0) {
4         return 1;
5     } else {
6         return n * factorial(n - 1);
7     }
8 }
9 int main() {
10     int n;
11     printf("Enter a non-negative integer: ");
12     scanf("%d", &n);
13
14     if (n < 0) {
15         printf("Factorial is not defined for negative numbers.\n");
16     } else {
17         unsigned long long result = factorial(n);
18         printf("Factorial of %d is %llu\n", n, result);
19     }
20     return 0;
21 }
```

Output:

```
/tmp/p4tpGPJwhZ.o
Enter a non-negative integer: 5
Factorial of 5 is 120

=== Code Execution Successful ===
```

5. Given a number n. the task is to print the Fibonacci series and the sum of the series using

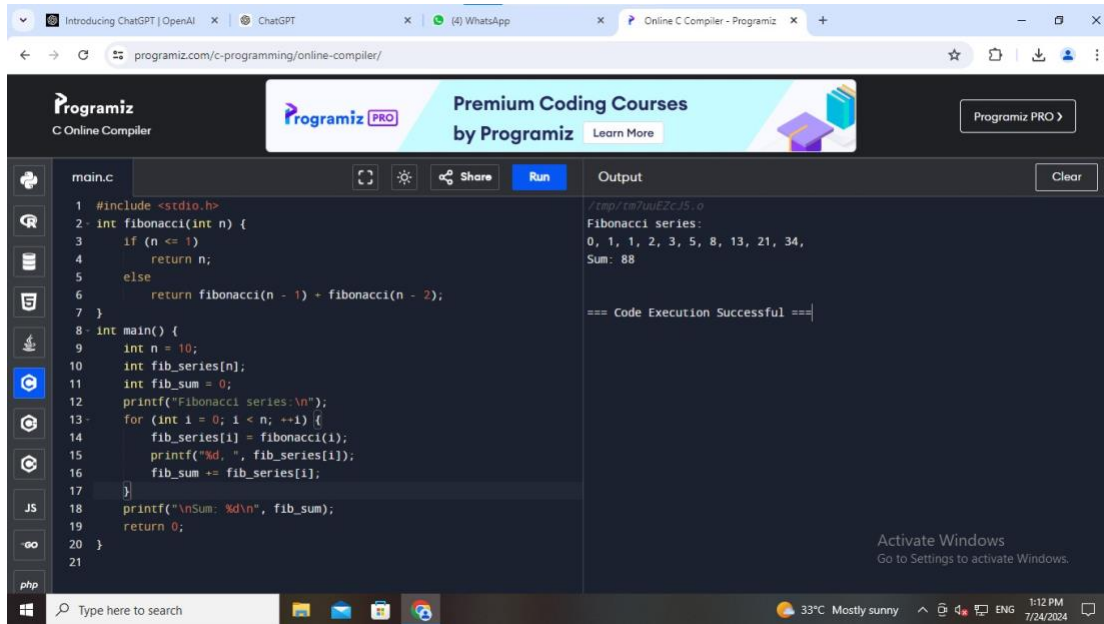
recursion.

input: n=10

output: Fibonacci series

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

Sum: 88



The screenshot shows the Programiz Online C Compiler interface. The code editor contains a C program that calculates the Fibonacci series for n=10 and prints the sum. The output window shows the Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and the sum: 88. The code is as follows:

```
main.c
1 #include <stdio.h>
2 int fibonacci(int n) {
3     if (n <= 1)
4         return n;
5     else
6         return fibonacci(n - 1) + fibonacci(n - 2);
7 }
8 int main() {
9     int n = 10;
10    int fib_series[n];
11    int fib_sum = 0;
12    printf("Fibonacci series:\n");
13    for (int i = 0; i < n; ++i) {
14        fib_series[i] = fibonacci(i);
15        printf("%d, ", fib_series[i]);
16        fib_sum += fib_series[i];
17    }
18    printf("\nSum: %d\n", fib_sum);
19    return 0;
20 }
21
```

Output:

```
/tmp/c7uuEzcJ5.o
Fibonacci series:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
Sum: 88

=== Code Execution Successful ===
```

6. You are given an array arr in increasing order. Find the element x from arr using binary

search.

Example 1: arr={ 1,5,6,7,9,10},X=6

Output : Element found at location 2

Example 2: arr={ 1,5,6,7,9,10},X=11

Output : Element not found at location 2

The screenshot displays the Programiz Online C Compiler web application. The browser's address bar shows the URL `programiz.com/c-programming/online-compiler/`. The interface includes a header with the Programiz logo, a navigation bar with a 'Run' button, and a sidebar with icons for various programming languages (C, C++, Java, JavaScript, PHP, Python, Ruby, Swift, Kotlin, Go, Rust, Rust, Swift, Kotlin, Go, Rust). The main editor area contains a C program named `main.c` that calculates the Fibonacci series for `n=10` and prints the sum. The output panel on the right shows the execution results: `Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, Sum: 88` and a confirmation message `=== Code Execution Successful ===`. The Windows taskbar at the bottom indicates the system time as 1:12 PM on 7/24/2024.

```
1 #include <stdio.h>
2 int fibonacci(int n) {
3     if (n <= 1)
4         return n;
5     else
6         return fibonacci(n - 1) + fibonacci(n - 2);
7 }
8 int main() {
9     int n = 10;
10    int fib_series[n];
11    int fib_sum = 0;
12    printf("Fibonacci series:\n");
13    for (int i = 0; i < n; ++i) {
14        fib_series[i] = fibonacci(i);
15        printf("%d, ", fib_series[i]);
16        fib_sum += fib_series[i];
17    }
18    printf("\nSum: %d\n", fib_sum);
19    return 0;
20 }
```

Output

```
/tmp/cc7uuEZcJS.o
Fibonacci series:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
Sum: 88

=== Code Execution Successful ===
```

7. You are given an array `arr` in increasing order. Find the element `x` from `arr` using linear search.

Example 1: `arr={ 1,5,6,7,9,10},X=6`

Output : Element found at location 2

Example 2: `arr={ 1,5,6,7,9,10},X=11`

Output : Element not found at location 2

Sign in to GitHub · GitHub

(8) WhatsApp

Introducing ChatGPT | OpenAI

ChatGPT

Online C Compiler - Programiz

programiz.com/c-programming/online-compiler/

Programiz

C Online Compiler

RADO

CAPTAIN COOK & TRUE SQUARE

DISCOVER NOW

Programiz PRO >

main.c

Run

Share

Clear

```
1 #include <stdio.h>
2 int linearSearch(int arr[], int n, int x) {
3     for (int i = 0; i < n; i++) {
4         if (arr[i] == x) {
5             return i + 1;
6         } else if (arr[i] > x) {
7             return -1;
8         }
9     }
10    return -1;
11 }
12 int main() {
13     int arr[] = {1, 5, 6, 7, 9, 10};
14     int n = sizeof(arr) / sizeof(arr[0]);
15     int x;
16     printf("Enter the element to search: ");
17     scanf("%d", &x);
18     int result = linearSearch(arr, n, x);
19     if (result != -1) {
20         printf("Element found at location %d\n", result);
21     } else {
22         printf("Element not found\n");
23     }
24     return 0;
25 }
26
```

Output

Enter the element to search: 7
Element found at location 4

=== Code Execution Successful ===

Type here to search

BSE smicap +1.82%

ENG

14:21

24-07-2024