Aim:

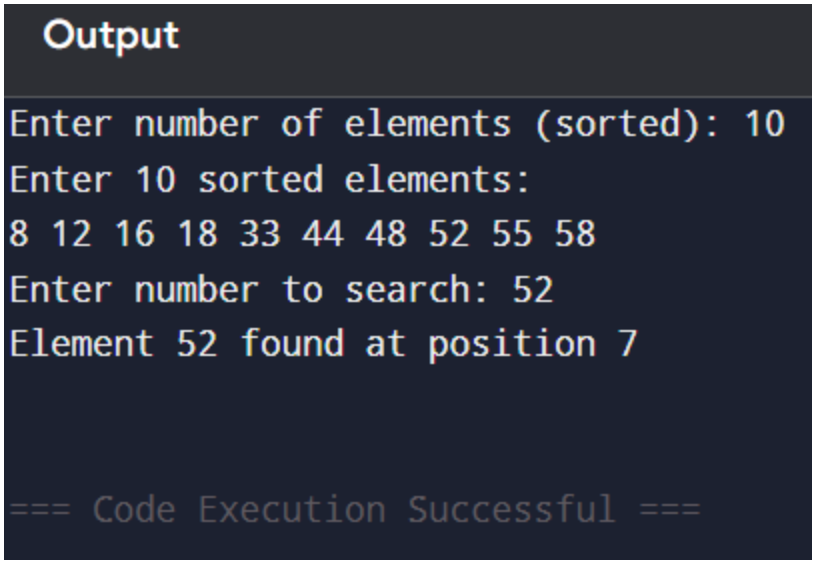
To implement **Binary Search** in C to find the position of a given element in a sorted array.

- **Algorithm:**

1. Start
2. Input a **sorted array** and the key to search.
3. Set `low = 0`, `high = n-1`.
4. Repeat until `low <= high`:
    - Find `mid = (low + high) / 2`.
    - If `arr[mid] == key`, return position.
    - If `arr[mid] > key`, set `high = mid - 1`.
    - Else set `low = mid + 1`.
5. If not found, display "Not Found".
6. Stop

CODE:

```
#include <stdio.h>

int main() {
   int arr[10], n, key, low, high, mid, found = 0;

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

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

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

   low = 0;
   high = n - 1;

   while (low <= high) {
      mid = (low + high) / 2;

      if (arr[mid] == key) {
         printf("Element %d found at position %d\n", key, mid);
         found = 1;
         break;
      }
      else if (arr[mid] > key)
         high = mid - 1;
      else
         low = mid + 1;
   }

   if (!found)
      printf("Element %d not found in array\n", key);

   return 0;
}
```



RESULT:

The program successfully executed and displayed the search element using binary search.