Program

```c
#include<stdio.h>
int binarySearch(int arr[], int n, int target)
 {
    int left=0,right=n-1,mid;
    while (left <= right)
     {
        mid = (left + right) / 2;

        if (arr[mid] == target)
        {
            return mid;
        }
        else if (arr[mid] < target)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }
    return -1;
}

int main()
{
    int i,n,target,result;
    printf("Enter number of elements :: ");
    scanf("%d",&n);
    int arr[n];
    printf("\nEnter %d elements in sorted order:: \n", n);
    for ( i = 0 ; i < n ; i++ )
    {
        scanf("%d",&arr[i]);
    }

    printf("\nEnter target value :");
    scanf("%d",&target);
    result = binarySearch(arr, n, target);

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

    return 0;
}
```

Experiment No: 19
Date:

BINARY SEARCH

Aim:

The aim of this program is to use the binary search method in C to find a specific number in a sorted list of numbers. If the number is found, the program will show its position in the list; if not, it will tell the user that the number is not in the list.

Algorithm:

1. Start
2. Create a function `BinarySearch(arr[], int n, int target)`.

   Set `left = 0`, `right = n - 1`, `mid`.

   Begin while loop: `left <= right`

   ```
   mid = (left + right) / 2
   if (arr[mid] == target)
       return mid
   else if (arr[mid] < target)
       left = mid + 1
   else
       right = mid - 1
   ```

   Return `-1`
3. Start the main function

   Initialize `i, n, target, result`.

   Input the number of elements from the user.

   Initialize `arr[n]`.

   Input the elements into the array.

   Ask the user to input the target value.

   Call the `binarySearch` function and set `result = binarySearch(arr, n, target)`.

   If `result != -1`

   Print: “Element found at location `result`.“

   Else

   Print: “Element not found.“
4. Stop

Output

```
Enter number of elements :: 5

Enter 5 elements in sorted order::
1
3
5
7
9

Enter target value :5
Element 5 found at index 2.


Enter number of elements :: 6

Enter 6 elements in sorted order::
10
20
30
40
50
60

Enter target value :25
Element 25 not found in the array.
```

Result:

The program has been executed successfully and obtained the output.