

Program 6:

Design, develop, code and run the program in any suitable language to implement the binary search algorithm. Determine the basis paths and using them derive different test cases, execute these test cases and discuss the test results.

Code:

```
#include<stdio.h>
int binsrc(int x[],int low, int high, int key)
{
    int mid;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(x[mid]==key)
            return mid;
        if(x[mid]<key)
            low=mid+1;
        else
            high=mid-1;
    }
    return -1;
}

int main()
{
    int a[20],key,i,n,succ;
    printf("Enter the n value");
    scanf("%d",&n);
    if(n>0)
    {
        printf("enter the elements in ascending order\n");
        for(i=0;i<n;i++)
            scanf("%d",&a[i]);

        printf("enter the key element to be searched\n");
        scanf("%d",&key);
        succ=binsrc(a,0,n-1,key);
        if(succ>=0)
```

```

printf("Element found in position = %d\n",succ+1);
else
printf("Element not found \n");
}
else
printf("Number of element should be greater than zero\n");
return 0;
}

```

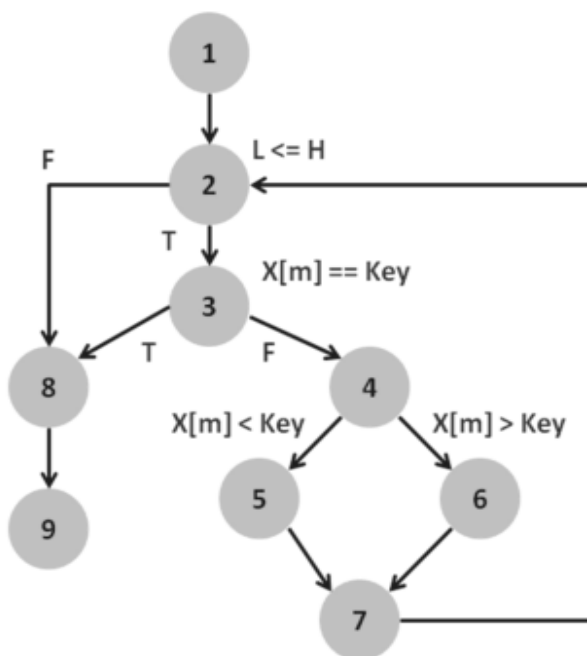
Binary Search function with line number

```

int binsrc(int x[],int low, int high, int key)
{
    int mid;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(x[mid]==key)
            return mid;
        if(x[mid]<key)
            low=mid+1;
        else
            high=mid-1;
    }
    return -1;
}

```

Program Graph – for Binary Search



Independent Paths:

#Edges=11, #Nodes=9, #P=1
 $V(G) = E - N + 2P = 11 - 9 + 2 = 4$

P1: 1-2-3-8-9

P2: 1-2-3-4-5-7-2

P3: 1-2-3-4-6-7-2

P4: 1-2-8-9

Pre-Conditions/Issues:

Array has Elements in Ascending order	T/F
Key element is in the Array	T/F
Array has ODD number of Elements	T/F

Test Cases – Binary Search

Paths	Inputs		Expected Output	Remarks
	X[]	Key		
P1: 1-2-3-8-9	{10,20,30,40,50}	30	Success	Key \in X[] and Key==X[mid]
P2: 1-2-3-4-5-7-2	{10,20,30,40,50}	20	Repeat and Success	Key < X[mid] Search 1 st Half
P3: 1-2-3-4-6-7-2	{10,20,30,40,50}	40	Repeat and Success	Key > X[mid] Search 2 nd Half
P4: 1-2-8-9	{10,20,30,40,50}	60 OR 05	Repeat and Failure	Key \notin X[]
P4: 1-2-8-9	Empty	Any Key	Failure	Empty List

PART B**Practical Based Learning**

Develop a Mini Project with documentation of suitable test-cases and their results to perform automation testing of any E-commerce or social media web page.

Suggested Guidelines:

- Create a WebDriver session.
- Navigate to a Web page.
- Locate the web elements on the navigated page.
- Perform an actions on the located elements.
- Assert the performed actions did the correct thing.
- Report the results of the assertions.
- End the session.

Each inputs / data feeds (ex: website, username, password, mobile no, product name, etc.,) must be provided through a file linked with code and neither to be entered manually nor to be included in the code

Note: Use any software testing tool like Selenium, Katalon, etc.