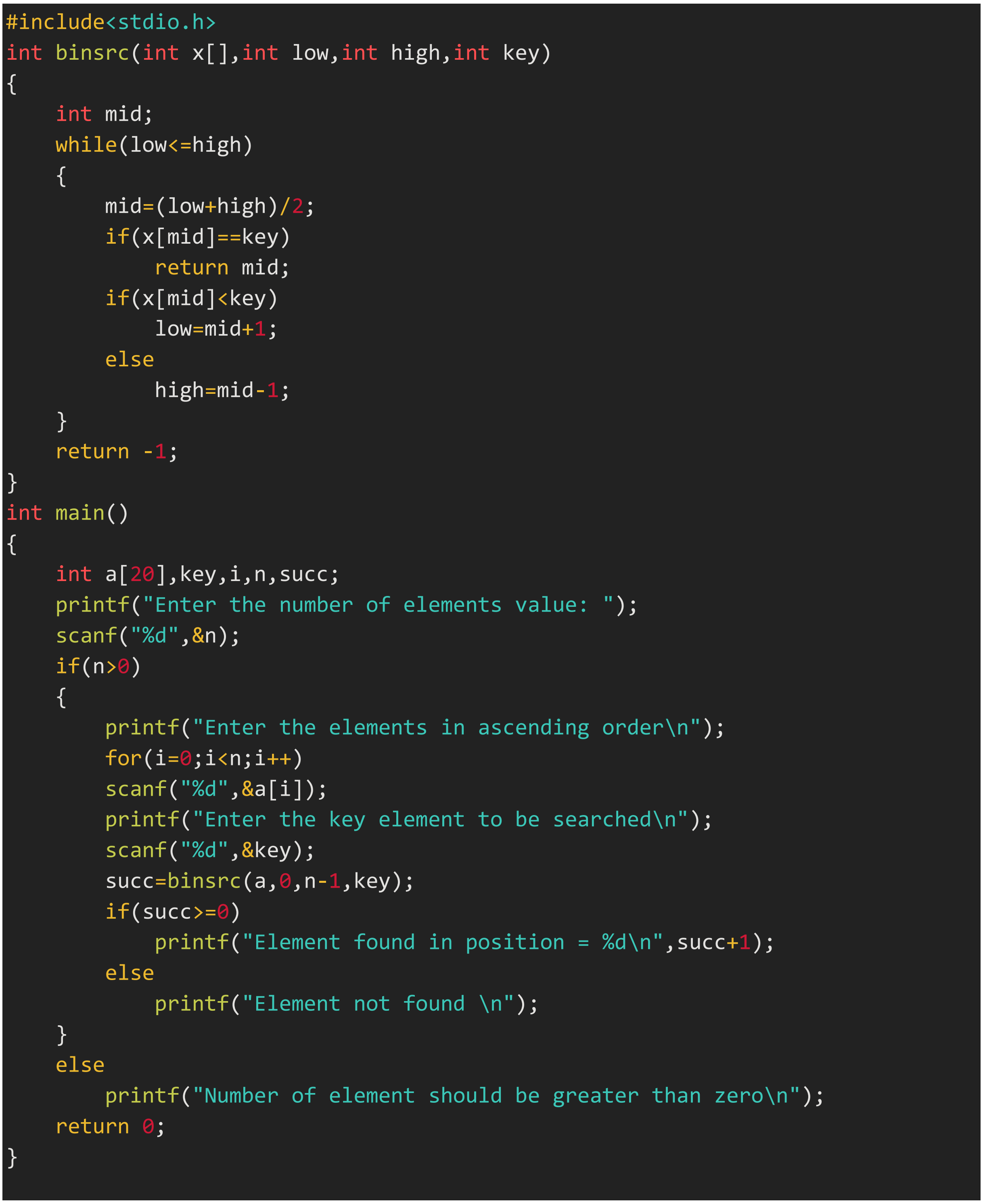
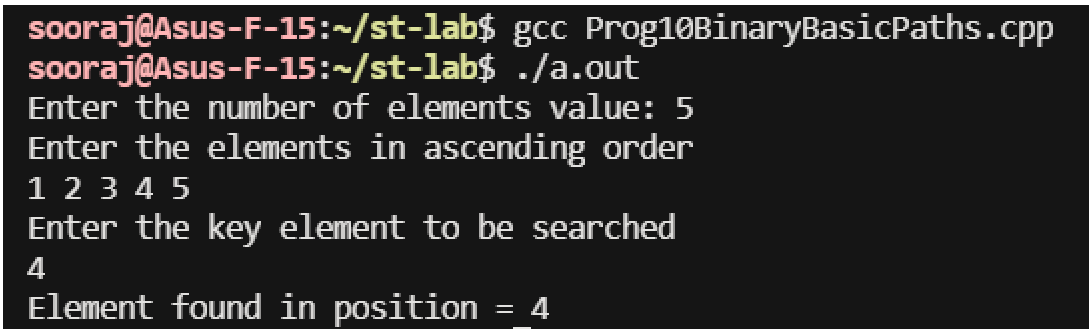
Program 10: Binary Search

Program:

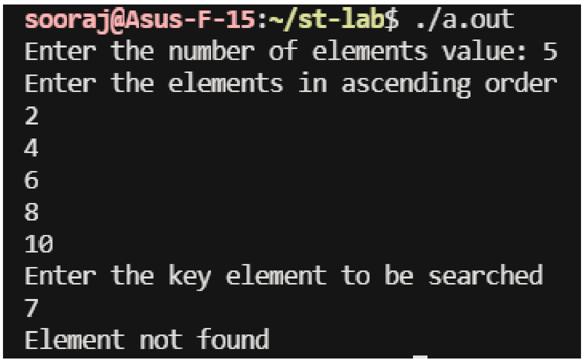


```c
#include<stdio.h>
int binsrc(int x[],int low,int high,int key)
{
    int mid;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(x[mid]==key)
            return mid;
        if(x[mid]<key)
            low=mid+1;
        else
            high=mid-1;
    }
    return -1;
}
int main()
{
    int a[20],key,i,n,succ;
    printf("Enter the number of elements 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;
}
```

Output:

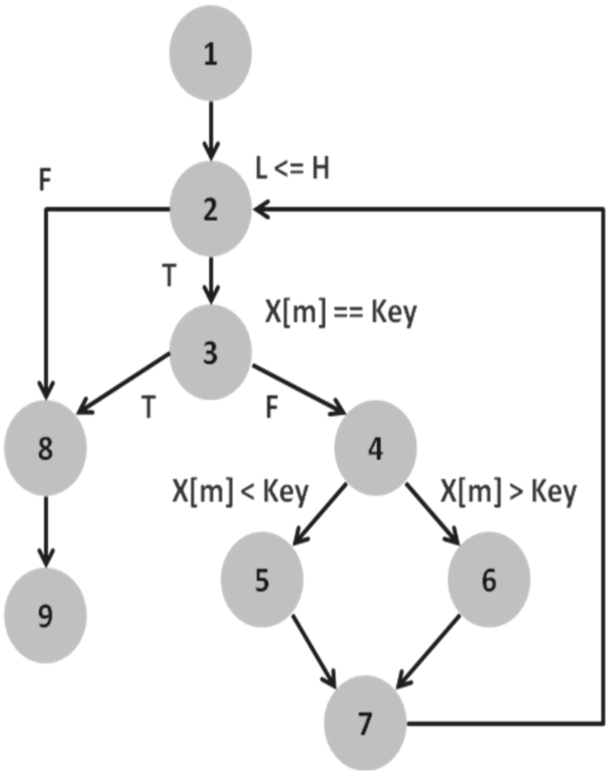


```
sooraj@Asus-F-15:~/st-lab$ gcc Prog10BinaryBasicPaths.cpp
sooraj@Asus-F-15:~/st-lab$ ./a.out
Enter the number of elements value: 5
Enter the elements in ascending order
1 2 3 4 5
Enter the key element to be searched
4
Element found in position = 4
```



```
sooraj@Asus-F-15:~/st-lab$ ./a.out
Enter the number of elements value: 5
Enter the elements in ascending order
2
4
6
8
10
Enter the key element to be searched
7
Element not found
```

Diagrams:



Independent Paths:

#Edges=11, #Nodes=9, #P=1

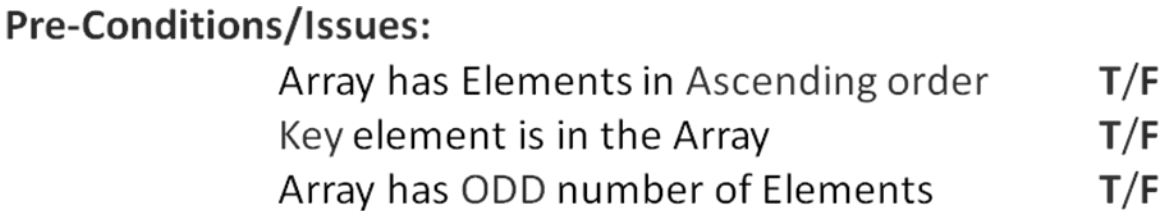
V(G)= E-N+2P = 11-9+2 = 4

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



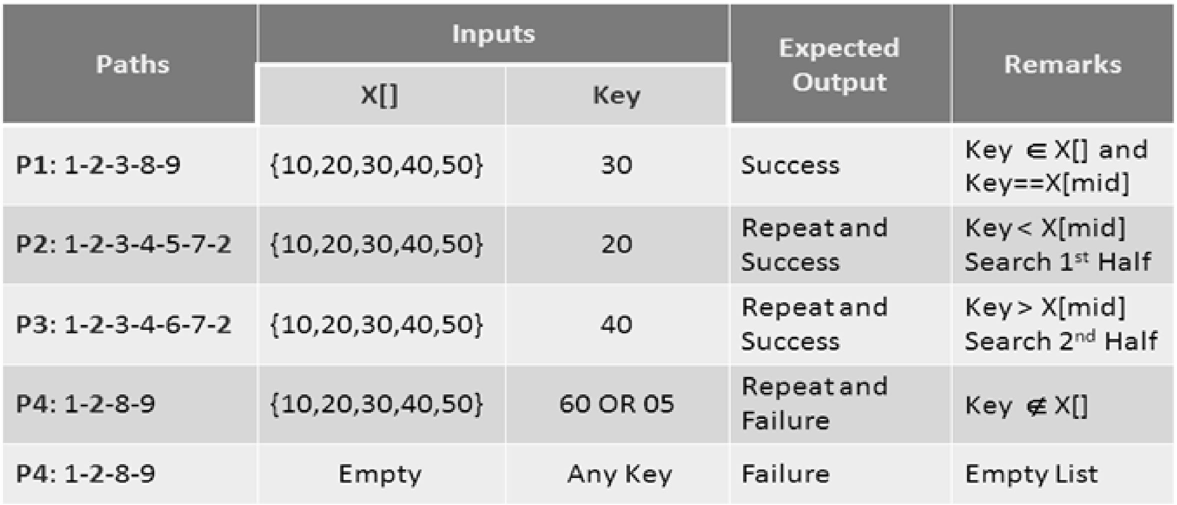
Pre-Conditions/Issues:

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 1st Half
P3: 1-2-3-4-6-7-2	{10,20,30,40,50}	40	Repeat and Success	Key > X[mid] Search 2nd 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