

# Samarth Programming Academy

---

## Binary Search Recursive Function

```
#include<stdio.h>
//Function Declaration
int Binary_Search(int[],int,int,int);
void Bubble_Sort(int[],int);
#define TRUE 1
#define FALSE 0
int main()
{
    int num = 0,value,Ret;
    printf("Enter how many element you want = ");
    scanf("%d",&num);

    //Filter
    if(num<=0)
    {
        printf("Invalid Size\n");
        return -1;
    }
    int arr[num]; //array creation

    //initialization of low and high variable
    int low = 0, high = num-1;

    //Accept Value from user

    for(int i=0;i<num;i++)
    {
        printf("Enter Number = ");
        scanf("%d",&arr[i]);
    }
```

## Samarth Programming Academy

---

### **//Display Array**

```
printf("Array = ");  
for(int i=0;i<num;i++)  
{  
    printf("%d ",arr[i]);  
}
```

### **//Accepting Value to Search**

```
printf("\nEnter Value to search = ");  
scanf("%d",&value);
```

### **//Function call to Bubble Sort to sort the array**

```
Bubble_Sort(arr,num);
```

### **//Function Call to Binary Search**

```
Ret = Binary_Search(arr,low,high,value);
```

```
if(Ret >= 0)  
{  
    printf("Value Found at %d index\n",Ret);  
}  
else  
{  
    printf("Value not found");  
}
```

```
return 0;
```

```
}
```

## Samarth Programming Academy

---

```
void Bubble_Sort(int arr[],int num)
{
    int temp,flag;
    for(int i=0;i<num-1;i++) //Loop for Passes i.e. N-1
    {
        flag = 0;
        for(int j=0;j<num-i-1;j++) //Loop for Comparison
        {
            if(arr[j] > arr[j+1])
            {
                //Swapping logic
                temp      = arr[j];
                arr[j]     = arr[j+1];
                arr[j+1]   = temp;
                flag = 1;
            }
        }
        if(flag==0)
        {
            break;
        }
    }
    return;
}
```

## Samarth Programming Academy

---

```
int Binary_Search(int arr[],int low,int high,int value)
{
    if(low>high)    //Base Condition
        return -1;

    int mid = (low+high)/2;

    if(value==arr[mid])
        return mid;

    else if(value>arr[mid])
        return Binary_Search(arr,mid+1,high,value);

    else if(value<arr[mid])
        return Binary_Search(arr,low,mid-1,value);

}
```