
Experiment 1.2

Student Name: Rajiv Paul

Branch: CSE

Semester: 3rd

Subject Name: Java Program Lab

UID: 20BCS1812

Section/Group: 6B

Date of Performance: 01/09/2021

Subject Code: 20CSP-219

Q1. Write a program in C/C++ that asks from user to enter any 10 array elements, and then ask to enter a number to search from the given array.

Aim/Overview of the practical:

To write a program to perform a searching operation.

2) Software required:

Vs Code

3) Source Code:

```
#include <iostream>
using namespace std;

int binarySearch(int arr[], int l, int r, int x)
{
    while (l <= r)
    {
        int m = l + (r - l) / 2;

        if (arr[m] == x)
            return m;

        if (arr[m] < x)
            l = m + 1;
        else
            r = m - 1;
    }

    return -1;
}
```

```
int main(void)
{
    int arr[10];

    cout << "Enter the 10 elements of array: " << endl;
    for (int i = 0; i < 10; ++i)
    {
        cout << "Enter the element of Index " << i << " : ";
        cin >> arr[i];
    }

    int x = -1;
    cout << "Enter the number you want to search for: " << endl;
    cin >> x;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = binarySearch(arr, 0, n - 1, x);
    cout << "\nElement is present at index " << result << " and the value is "
        << x << endl;
    return 0;
}
```

Caption

4. Output:

```
Enter the 10 elements of array:
Enter the element of Index 0 : 1
Enter the element of Index 1 : 2
Enter the element of Index 2 : 3
Enter the element of Index 3 : 4
Enter the element of Index 4 : 5
Enter the element of Index 5 : 6
Enter the element of Index 6 : 7
Enter the element of Index 7 : 8
Enter the element of Index 8 : 9
Enter the element of Index 9 : 0
Enter the number you want to search for:
3

Element is present at index 2 and the value is 3
```

Q2. Write a program to insert and delete values from a binary search.

Aim/Overview of the practical:

To write a program to perform a insertion and deletion operation.

2) Software required:

Vs Code

3) Source Code:

Insertion:

```
#include <iostream>
using namespace std;

// Insertion
int main()
{
    int size;
    cout << "Enter the size of array: ";
    cin >> size;
    int arr[size];
    for (int i = 0; i < size; i++) {
        cout << "Index of the array " << i << " : ";
        cin >> arr[i];
    }

    int n;
    cout << "Enter the value you want to insert" << endl;
    cin >> n;
    int i;
    for(i=size-1; (i >= 0 && arr[i] > n); i--)
    {
        arr[i+1] = arr[i];
    }
    arr[i+1]=n;
    size = size+1;
    cout << "Array after insertion" << endl;
    for(int i=0; i<size; i++) {
        cout << arr[i] << " ";
    }
    cout<<endl;
    return 0;
}
```

Deletion:

```
#include <iostream>
using namespace std;
int main()
{
    int size;
    cout << "Enter the size of array: ";
    cin >> size;
    int arr[size];
    for (int i = 0; i < size; i++)
    {
        cout << "Index of the array " << i << " : ";
        cin >> arr[i];
    }

    int n;
    cout << "Enter the value you want to delete" << endl;
    cin >> n;
    int i, loc;
    for(i=0; i<size; i++)
    {
        if(arr[i] == n)
        {
            loc = i;
        }
    }
    size = size-1;
    for(i=loc; i<size; i++)
    {
        arr[i] = arr[i+1];
    }
    cout << "Array after deletion" << endl;
    for(int i=0; i<size; i++) {
        cout << arr[i] << " ";
    }
    cout<<endl;

    return 0;
}
```

4. Output:

Insertion:

```
Enter the size of array: 2
Index of the array 0 : 1
Index of the array 1 : 2
Enter the value you want to insert
09
Array after insertion
1 2 9
```

Deletion:

```
Enter the size of array: 3
Index of the array 0 : 1
Index of the array 1 : 2
Index of the array 2 : 9
Enter the value you want to delete
2
Array after deletion
1 9
```

Q2. Write a program to count the number of nodes in a binary search.

Aim/Overview of the practical:

To write a program to count the number of node in a binary search .

2) Software required:

Vs Code

3) Source Code:

```
#include<iostream>
using namespace std;

int n=1;

struct node
{
    int data;
    node* left;
    node* right;
};

struct node* getNode(int data)
{
    node* newNode=new node();
    newNode->data=data;
    newNode->left=NULL;
    newNode->right=NULL;
    return newNode;
}

struct node* Insert(struct node* root, int data)
{
    if (root == NULL)
        return getNode(data);

    if (data < root->data)
        root->left = Insert(root->left, data);
    else if (data > root->data)
        root->right = Insert(root->right, data);
```

```
    return root;
}

int CountNodes(node*root)
{
    if(root==NULL)
        return 0;
    if(root->left!=NULL)
    {
        n=n+1;
        n=CountNodes(root->left);
    }
    if(root->right!=NULL)
    {
        n=n+1;
        n=CountNodes(root->right);
    }
    return n;
}
```

```
int main()
{
    node* root=NULL;
    root=Insert(root,3);
    Insert(root,4);
    Insert(root,2);
    Insert(root,5);
    Insert(root,1);

    cout<<"\nTotal No. of Nodes in the Binary Search Tree = "<<CountNodes(root)
    <<endl<<endl;

    return 0;
}
```

4. Output:

```
Total No. of Nodes in the Binary Search Tree = 5
```