

Ex. No. 7	IMPLEMENTATION OF LINEAR SEARCH AND BINARY SEARCH
30-08-2017	

Question:

To implement the working of implementation of linear search and binary search.

Algorithm:

1. Start.
2. For linear search, follow the steps:
3. Create global variable int i, key.
4. Inside the main function for(i=0; i<5;i++)

{cin>>A[i]; }
5. Followed by for(i=0; i<5;i++)
6. Inside the loop, check if(A[i]==key)
7. If the condition is true, we will print the message as key is found.
8. Else, we will print the message as key is found.
9. For binary search, follow the steps:
10. Initialize left as 0 and right as 5.
11. Initialize a while loop with condition left<=right.
12. Calculate mid as (left + right)/2.
13. Check if a[mid]=data ,if true then display element found.
14. Else if(a[mid]>data),then update right as mid-1.
15. Else if(a[mid]<data),then update left as mid+1.
16. End

Program:

/*To implement the working of implementation of linear search and binary search.*/

```
#include <iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
void lsearch(int a[5],int data)
```

```
{
```

```
    for(int i=0;i<5;i++)
```

```
    {
```

```
        if (a[i]==data)
```

```
        {
```

```
            cout<<"\n Data "<<data<<" found at position "<<i<<endl;
```

```
            return;
```

```
        }
```

```
    }
```

```
    cout<<"\n Data to be searched not found"<<endl;
```

```
}
```

```
void bsearch(int a[5],int data)
```

```
{
```

```
    int left =0;
```

```
int right=5;

while(left<=right)
{
    int mid=(left + right)/2;

    if (a[mid]==data)
    {
        cout<<"\n Data "<<data<<" found at "<<mid<<endl;
        return;
    }
    else if(a[mid]>data)
    {
        right=mid-1;
    }
    else if(a[mid]<data)
    {
        left=mid+1;
    }
}

cout<<"\n Data not found"<<endl;
}
```

```
int main()
{
    int opt,dat;
    int a[5];
    while(1)
    {
        cout<<"\n Choices to do ";
        cout<<"\n-----";
        cout<<"\n 1.Linear search";
        cout<<"\n 2.Binary search";
        cout<<"\n 3.Exit";
        cout<<"\n\n Enter  your choice: ";
        cin>>opt;
        switch(opt)
        {
            case 1:
                cout<<"\n Enter  the array elements: ";
                for(int i=0;i<5;i++)
                {
                    cin>>a[i];
                }
            
```

```
    cout<<"\n Enter the data to be searched: ";

    cin>>dat;

    lsearch(a,dat);

    break;

case 2:

    cout<<"\n Enter the array elements: ";

    for(int i=0;i<5;i++)

    {

        cin>>a[i];

    }

    cout<<"\n Enter the data to be searched: ";

    cin>>dat;

    bsearch(a, dat);

    break;

case 3:

    exit(0);

default:

    cout<<"\n Sorry wrong option"<<endl;

    break;

}

}
```

return 0;}

DATA STRUCTURES LAB

Output:

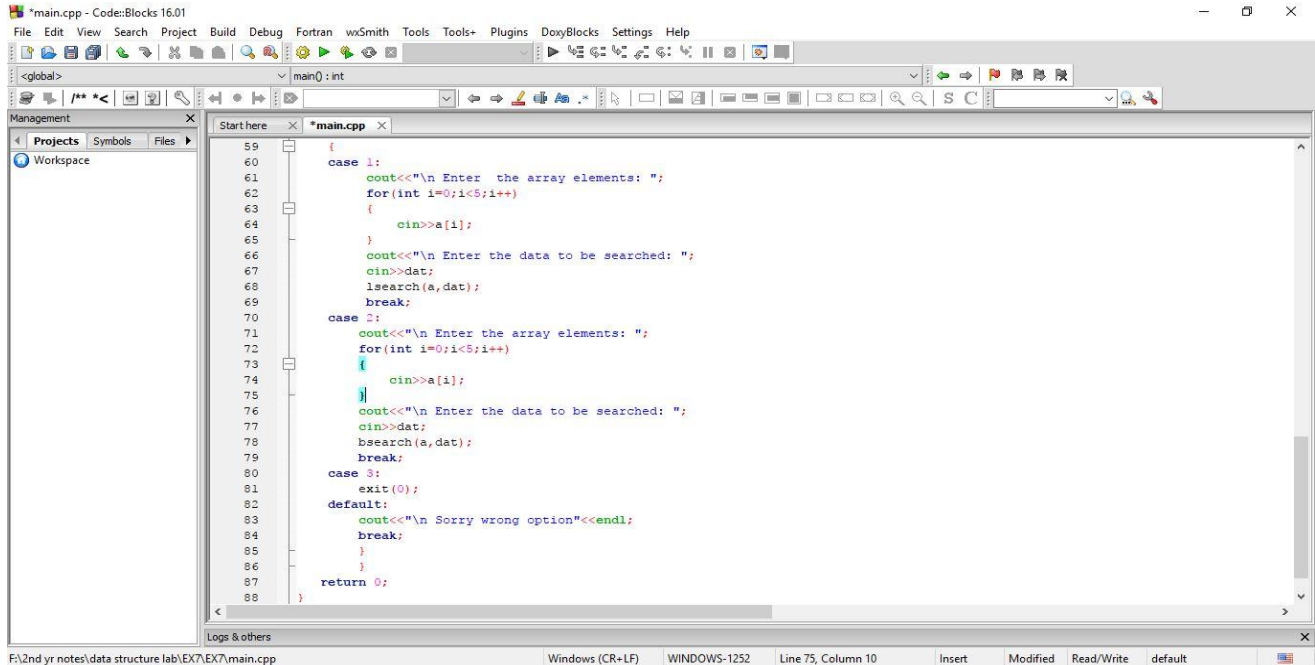
The screenshot shows the Code::Blocks IDE with the file `*main.cpp` open. The code implements a linear search function. The `lsearch` function takes an array `a` of size 5 and a `data` value. It iterates through the array from index 0 to 4. If the current element matches the `data`, it prints the data and its position, then returns. If no match is found, it prints a message indicating the data was not found. The `bsearch` function is also present but not yet implemented.

```
1  /*To implement the working of implementation of linear search and binary search.*/
2  #include <iostream>
3  #include<stdlib.h>
4  using namespace std;
5
6  void lsearch(int a[5],int data)
7  {
8      for(int i=0;i<5;i++)
9      {
10         if (a[i]==data)
11         {
12             cout<<"\n Data '"<<data<<"' found at position "<<i<<endl;
13             return;
14         }
15     }
16     cout<<"\n Data to be searched not found"<<endl;
17 }
18
19 void bsearch(int a[5],int data)
20 {
21     int left =0;
22     int right=5;
23
24     while(left<right)
25     {
26         int mid=(left+right)/2;
27
28         if (a[mid]==data)
29         {
30             cout<<"\n Data '"<<data<<"' found at "<<mid<<endl;
```

The screenshot shows the Code::Blocks IDE with the file `*main.cpp` open. The code continues from the previous screenshot, implementing the `bsearch` function and the `main` function. The `bsearch` function uses a while loop to find the middle element and compares it with the `data`. If the middle element is greater than the `data`, it updates the `right` pointer. If it is less, it updates the `left` pointer. If the `data` is not found, it prints a message. The `main` function prompts the user to choose between linear search, binary search, or exit, and then calls the appropriate function based on the choice.

```
30     cout<<"\n Data '"<<data<<"' found at "<<mid<<endl;
31     return;
32 }
33 else if(a[mid]>data)
34 {
35     right=mid-1;
36 }
37 else if(a[mid]<data)
38 {
39     left=mid+1;
40 }
41 }
42 cout<<"\n Data not found"<<endl;
43 }
44
45 int main()
46 {
47     int opt,dat;
48     int a[5];
49     while(1)
50     {
51         cout<<"\n Choices to do ";
52         cout<<"\n-----";
53         cout<<"\n 1.Linear search";
54         cout<<"\n 2.Binary search";
55         cout<<"\n 3.Exit";
56         cout<<"\n\n Enter your choice: ";
57         cin>>opt;
58         switch(opt)
59         {
```

DATA STRUCTURES LAB



```
59 {
60     case 1:
61         cout<<"\n Enter the array elements: ";
62         for(int i=0;i<5;i++)
63         {
64             cin>>a[i];
65         }
66         cout<<"\n Enter the data to be searched: ";
67         cin>>dat;
68         lsearch(a,dat);
69         break;
70     case 2:
71         cout<<"\n Enter the array elements: ";
72         for(int i=0;i<5;i++)
73         {
74             cin>>a[i];
75         }
76         cout<<"\n Enter the data to be searched: ";
77         cin>>dat;
78         bsearch(a,dat);
79         break;
80     case 3:
81         exit(0);
82     default:
83         cout<<"\n Sorry wrong option"<<endl;
84         break;
85     }
86 }
87 return 0;
88 }
```

"F:\2nd yr notes\data structure lab\EX7\EX7\main.exe"

Choices to do

- 1.Linear search
- 2.Binary search
- 3.Exit

Enter your choice: 1

Enter the array elements: 1 2 3 4 5

Enter the data to be searched: 4

Data '4' found at position 3

Choices to do

- 1.Linear search
- 2.Binary search
- 3.Exit

Enter your choice: 1

Enter the array elements: 1 3 7 6 2

Enter the data to be searched: 10

Data to be searched not found

"F:\2nd yr notes\data structure lab\EX7\EX7\main.exe"

Choices to do

- 1.Linear search
- 2.Binary search
- 3.Exit

Enter your choice: 2

Enter the array elements: 1 2 3 4 5

Enter the data to be searched: 4

Data '4' found at 3

Choices to do

- 1.Linear search
- 2.Binary search
- 3.Exit

Enter your choice: 2

Enter the array elements: 1 2 3 4 5

Enter the data to be searched: 7

Data '7' found at 5

Choices to do

- 1.Linear search
- 2.Binary search
- 3.Exit

Enter your choice: 3

VIDEO URL:

<https://youtu.be/ZOifOjiQOwY>

RESULT:

The program of implementation of linear and binary search is implemented successfully and the output is verified.