Lab 07

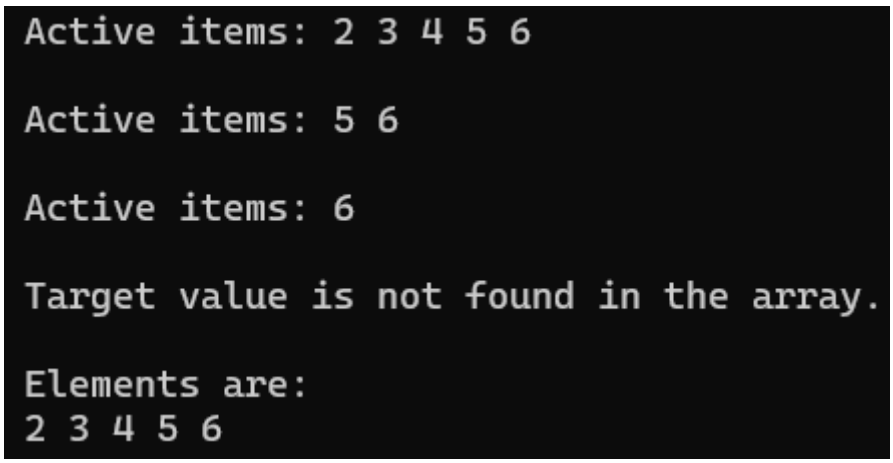
Given a sorted array of n integers and a target value, find the index of the target value in the array. Show all active items [divided half] at every stage. Handle the situation when the target value is not found.

```
#include <iostream>
using namespace std;
struct Array
{
   int A[10];
   int size;
   int length;
};
void Display(struct Array arr)
{
   cout << "\n Elements are: " << endl;
   for (int i = 0; i < arr.length; i++)
   {
      cout << " " << arr.A[i];
   }
   cout << endl;
}
void DisplayActiveItems(struct Array arr, int l, int h)
{
   cout << "\n Active items: ";
   for (int i = l; i <= h; i++)
```

```
    {
        cout << arr.A[i] << " ";
    }
    cout << endl;
}
int BinarySearch(struct Array arr, int target)
{
    int l = 0, h = arr.length - 1;
    while (l <= h)
    {
        int mid = (l + h) / 2;
        DisplayActiveItems(arr, l, h);
        if (target == arr.A[mid])
            return mid;
        else if (target < arr.A[mid])
            h = mid - 1;
        else
            l = mid + 1;
    }
    return -1;
}
int main()
{
    struct Array arr = { {2, 3, 4, 5, 6}, 10, 5 };
    int target = 10;
    int result = BinarySearch(arr, target);
    if (result != -1)
    {
        cout << "\n Target value is found at index " << result << "." << endl;
```

```
    }
    else
    {
        cout << "\n Target value is not found in the array." << endl;
    }
    Display(arr);
    return 0;
}
```

Output



Given a sorted array of n integers and a target value, find the first occurrence of the target value in the array.

```
#include <iostream>
using namespace std;
struct Array
{
    int A[10];
    int size;
    int length;
```

```
};
void Display(struct Array arr)
{
   cout << "\n Elements are: " << endl;
   for (int i = 0; i < arr.length; i++)
   {
      cout << " " << arr.A[i];
   }
   cout << endl;
}
int BinarySearch(struct Array arr, int target)
{
   int l = 0, h = arr.length - 1, result = -1;
   while (l <= h)
   {
      int mid = (l + h) / 2;
      if (target == arr.A[mid])
      {
         result = mid;
         h = mid - 1;//continue searching on the left hand side for the
occurrence
      }
      else if (target < arr.A[mid])
      {
         h = mid - 1;
      }
      else
      {
         l = mid + 1;
```

```
        }
    }
    return result;
}
int main()
{
    struct Array arr = { {2, 3, 4, 4, 4, 5, 6}, 10, 7 };
    int target = 4;
    int result = BinarySearch(arr, target);
    if (result != -1)
    {
        cout << "\n Target value is found at index " << result << "." << endl;
    }
    else
    {
        cout << "\n Target value is not found in the array." << endl;
    }
    Display(arr);
    return 0;
}
```

Output

```
Target value is found at index 2.

Elements are:
2 3 4 4 4 5 6
```

Given a sorted array of n integers and a target value, find the last occurrence of the target value in the array.

```
#include <iostream>
using namespace std;
struct Array
{
   int A[10];
   int size;
   int length;
};
void Display(struct Array arr)
{
   cout << "\n Elements are: " << endl;
   for (int i = 0; i < arr.length; i++)
   {
      cout << " " << arr.A[i];
   }
   cout << endl;
}
int BinarySearch(struct Array arr, int target)
{
   int l = 0, h = arr.length - 1, result = -1;
   while (l <= h)
   {
      int mid = (l + h) / 2;
      if (target == arr.A[mid])
      {
         result = mid;
         l = mid + 1;//continue searching on the right hand side for the
occurrence
      }
```

```
        else if (target < arr.A[mid])
        {
            h = mid - 1;
        }
        else
        {
            l = mid + 1;
        }
    }
    return result;
}
int main()
{
    struct Array arr = { {2, 3, 4, 4, 4, 5, 6}, 10, 7 };
    int target = 4;
    int result = BinarySearch(arr, target);
    if (result != -1)
    {
        cout << "\n Target value is found at index " << result << "." << endl;
    }
    else
    {
        cout << "\n Target value is not found in the array." << endl;
    }
    Display(arr);
    return 0;
}
```

Output

```
Target value is found at index 4.

Elements are:
2 3 4 4 4 5 6
```

Given a sorted array of n integers and a target value, find the number of occurrences of the target value in the array.

```
#include <iostream>
using namespace std;
struct Array
{
   int A[10];
   int size;
   int length;
};
void Display(struct Array arr)
{
   cout << "\n Elements are: " << endl;
   for (int i = 0; i < arr.length; i++)
   {
      cout << " " << arr.A[i];
   }
   cout << endl;
}
int FirstOccurrence(struct Array arr, int target)
{
   int l = 0, h = arr.length - 1, result = -1;
```

```
    while (l <= h)
    {
        int mid = (l + h) / 2;
        if (target == arr.A[mid])
        {
            result = mid;
            h = mid - 1;//continue searching on the left hand side for the
occurrence
        }
        else if (target < arr.A[mid])
        {
            h = mid - 1;
        }
        else
        {
            l = mid + 1;
        }
    }
    return result;
}
int LastOccurrence(struct Array arr, int target)
{
    int l = 0, h = arr.length - 1, result = -1;
    while (l <= h)
    {
        int mid = (l + h) / 2;
        if (target == arr.A[mid])
        {
            result = mid;
```

```
            l = mid + 1;//continue searching on the right hand side for the
occurrence
        }
        else if (target < arr.A[mid])
        {
            h = mid - 1;
        }
        else
        {
            l = mid + 1;
        }
    }
    return result;
}
int NumberofOccurrences(struct Array arr, int target)
{
    int FirstIndex = FirstOccurrence(arr, target);
    if (FirstIndex == -1)
    {
        return 0;
    }
    int LastIndex = LastOccurrence(arr, target);
    return (LastIndex - FirstIndex + 1);
}
int main()
{
    struct Array arr = { {2, 3, 4, 4, 4, 5, 6}, 10, 7 };
    int target = 4;
    int count = NumberofOccurrences(arr, target);
```

```
    if (count > 0)
    {
        cout << "\n The target value " << target << " occurs " << count << "
times in the array." << endl;
    }
    else
    {
        cout << "\n Target value is not found in the array." << endl;
    }
    Display(arr);
    return 0;
}
```

Output

```
The target value 4 occurs 3 times in the array.

Elements are:
2 3 4 4 4 5 6
```
