Lab Report 02

Submitted by

Muhammad Awais / 2883

Submitted To

Ma’am Zanaib Malik

Department of Computer Science and Engineering

National University of modern language, Islamabad

List of Task

1. Setup our concepts of Array.
2. Find largest value location in arry.
3. Swap Function in arry.
4. Linear Search
5. Binary Search
6. Selection sort
7. Bubble sort
8. Insertion sort as a **home Task**

Description of int largest location

```
/*write a Function name LargestLoc  that will an arry,
its size  as parametrs, this function should display the Index of largest in the
Arry*/
int largestLoc(int arr[], int size);// Largest loction Finder
```

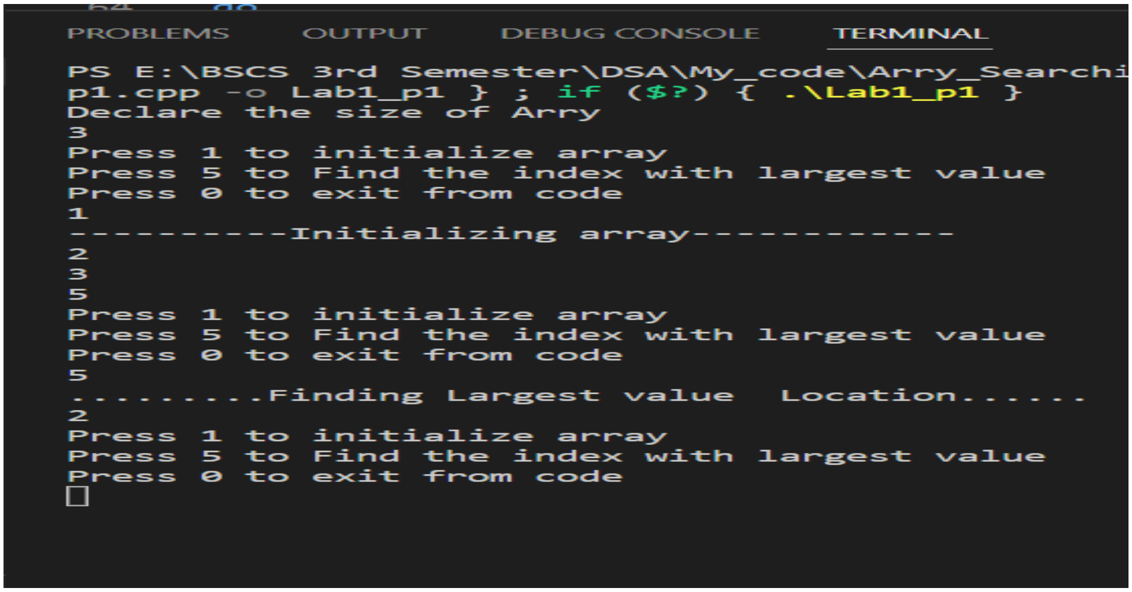
What does Function Do?

1. Int function means **will return.**
2. Take an array and size of array as a parameter.
3. First initialize a variable from zero index of arry largest=arr[0];
4. Compare the values of array with largest if array element if Greater than largest then assign the value of array element to largest and assign index of that element to largestloc and **return largestloc.**
5. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code

```
int largestLoc(int arr[], int size)
{
    int largest = arr[0];
    int largestloc=0;
    for (int i = 0; i < size; i++)
    {
        if(arr[i]>largest)
        {
            largest=arr[i];
            largestloc=i;
        }
    }
      return largestloc;

}// End of largestLoc
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searchi
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
3
Press 1 to initialize array
Press 5 to Find the index with largest value
Press 0 to exit from code
1
----------Initializing array------------
2
3
5
Press 1 to initialize array
Press 5 to Find the index with largest value
Press 0 to exit from code
5
.........Finding Largest value  Location......
2
Press 1 to initialize array
Press 5 to Find the index with largest value
Press 0 to exit from code
□
```

Description of void Swap

```
/*write a Function name Swap  that will an arry,
and two indices   as parametrs, this function swap the indices of Arry*/
void Swap(int arr[], int indexA,int indexB);// Swap indices
```

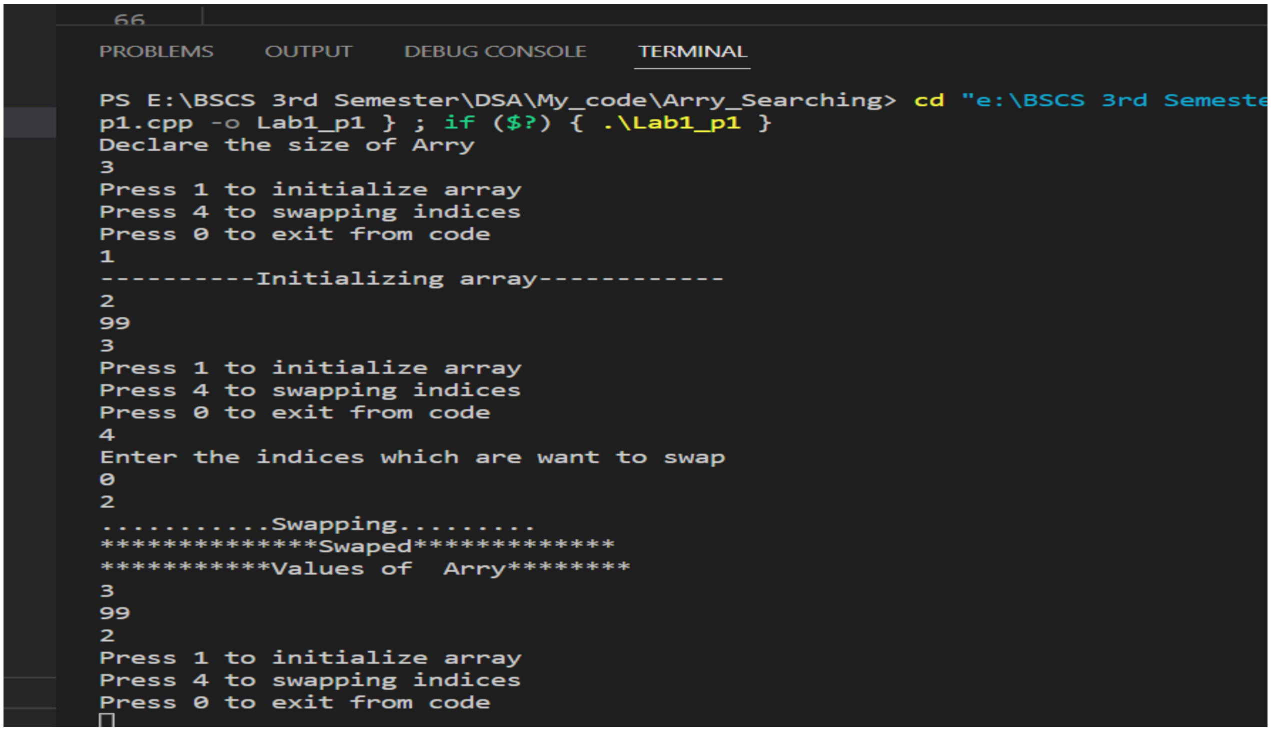
What does Function Do?

1. Void function means **nothing will return.**
2. Take an array and indices of array as a parameter.
3. Swap indices by using another variable temp, which store first indexX then index store IndexY then indexY store temp.
4. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code

```
void Swap(int arr[], int indexA, int indexB)
{
    int temp;
    temp=arr[indexA];
    arr[indexA]=arr[indexB];
    arr[indexB]=temp;

}// End of Swap
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Semeste
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
3
Press 1 to initialize array
Press 4 to swapping indices
Press 0 to exit from code
1
----------Initializing array------------
2
99
3
Press 1 to initialize array
Press 4 to swapping indices
Press 0 to exit from code
4
Enter the indices which are want to swap
0
2
...........Swapping.........
**************Swaped*************
***********Values of  Arry********
3
99
2
Press 1 to initialize array
Press 4 to swapping indices
Press 0 to exit from code
```

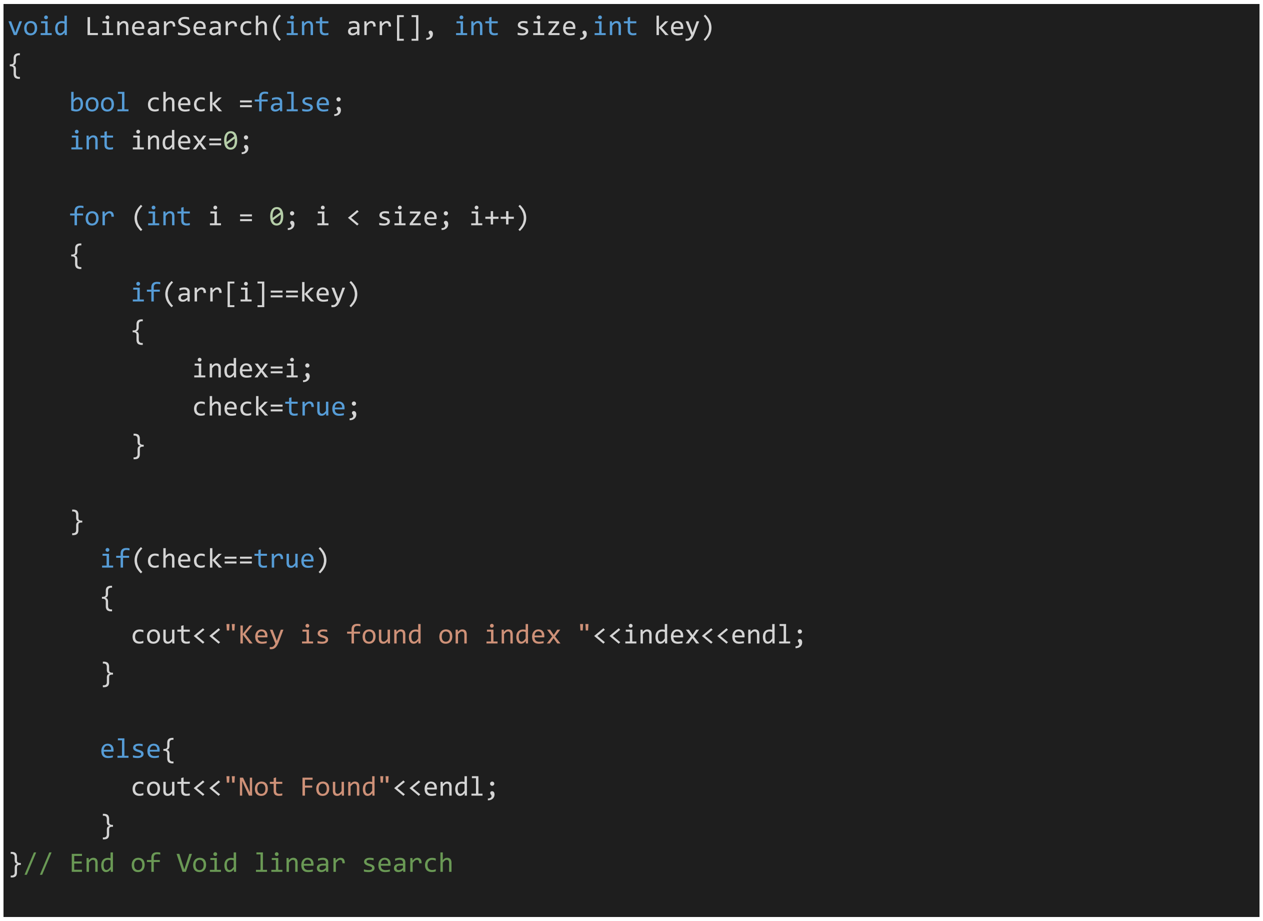
Description of void linear Search

```
/*write a Function name Binary search  that will an arry,
size of arry and key to find  as parametrs, this function search a entered key in
the Arry*/
void LinearSearch(int arr[], int size,int key);// linear Search
```

What does Function Do?

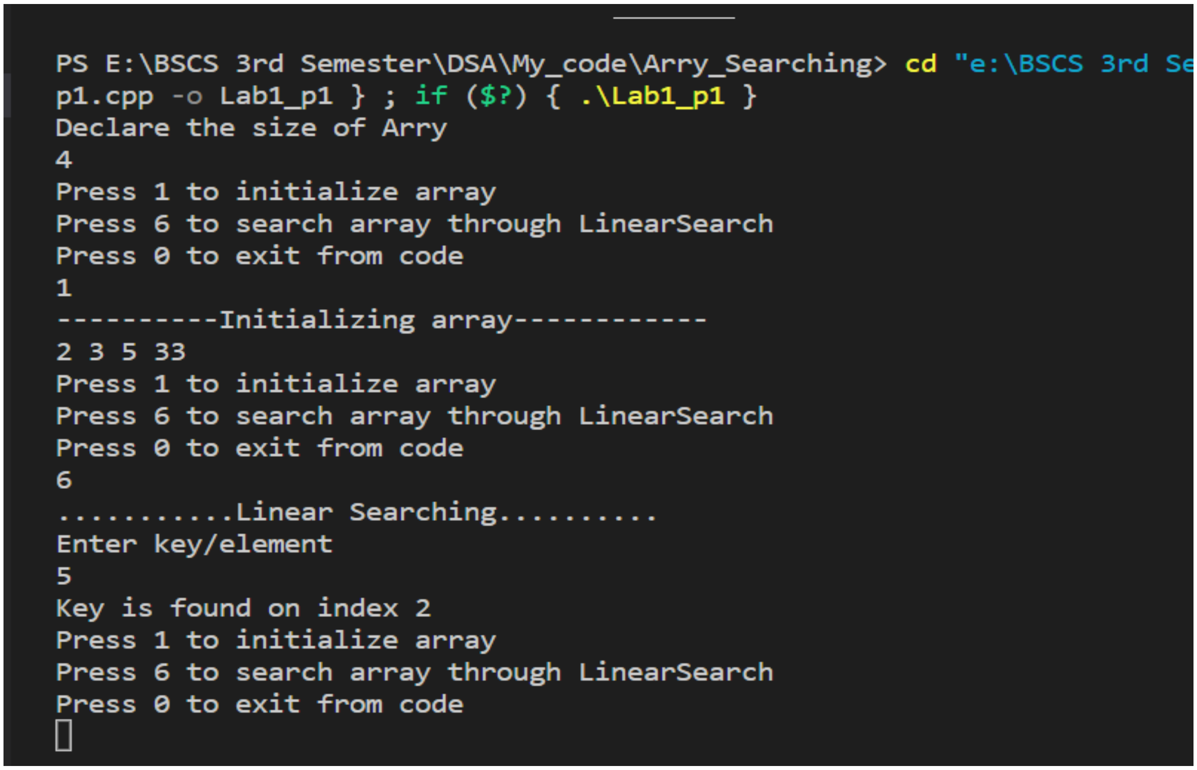
1. void function means **nothing will return.**
2. Take an array and size of array as a parameter.
3. Search element in array using linear search method.
4. Use Boolean to check whether the element found or not.
5. Use loop to traverse the array.
6. Use condition if array of element is equal to key return index and bool= true.
7. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code



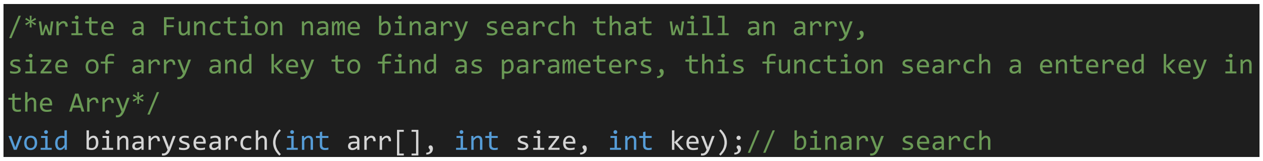
```
void LinearSearch(int arr[], int size,int key)
{
    bool check =false;
    int index=0;

    for (int i = 0; i < size; i++)
    {
        if(arr[i]==key)
        {
            index=i;
            check=true;
        }

    }
      if(check==true)
      {
        cout<<"Key is found on index "<<index<<endl;
      }

      else{
        cout<<"Not Found"<<endl;
      }
}// End of Void linear search
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Se
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
4
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 0 to exit from code
1
----------Initializing array------------
2 3 5 33
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 0 to exit from code
6
...........Linear Searching..........
Enter key/element
5
Key is found on index 2
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 0 to exit from code
```

Description of void Binary Search



```
/*write a Function name binary search that will an arry,
size of arry and key to find as parameters, this function search a entered key in
the Arry*/
void binarysearch(int arr[], int size, int key);// binary search
```

What does Function Do?

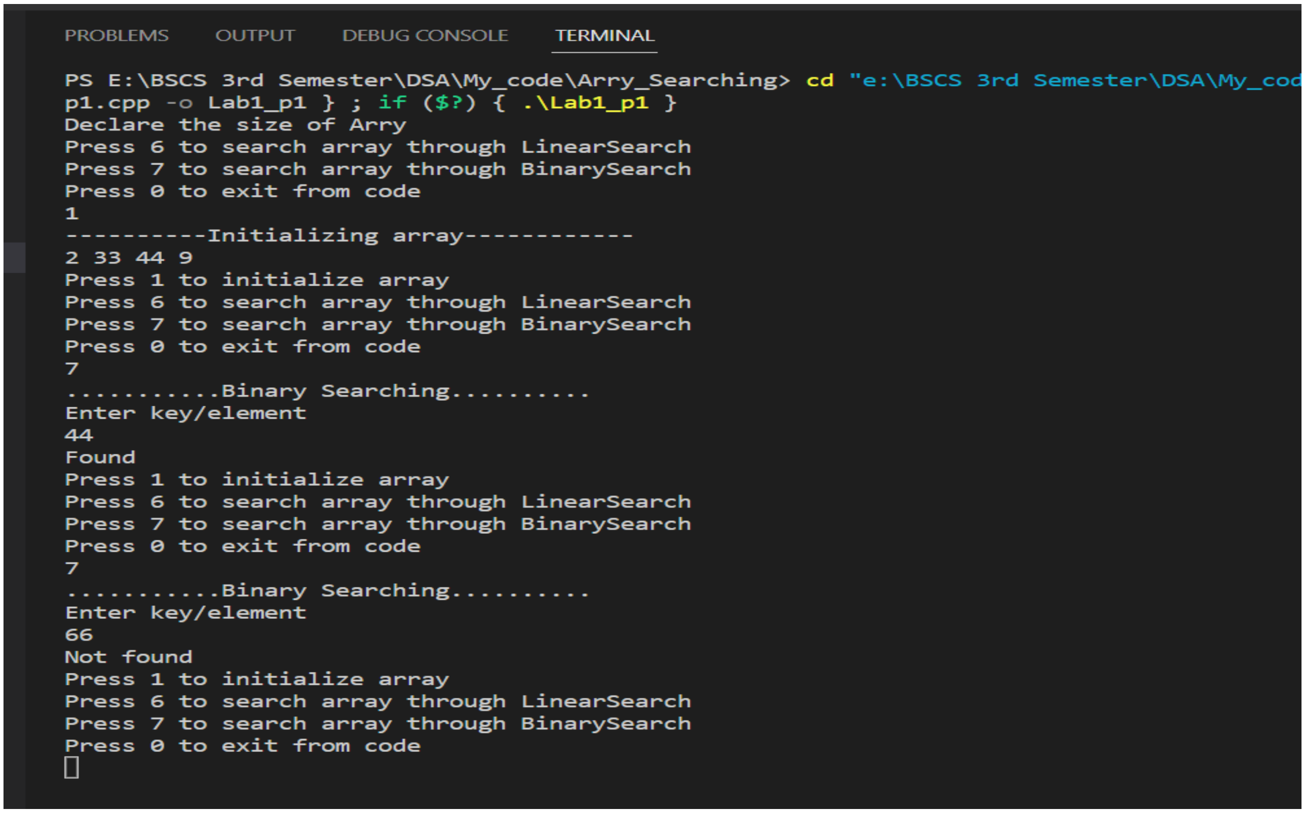
1. void function means **nothing will return.**
2. Take an array and size of array as a parameter.
3. Search element in array using binary search method.
4. We need a starting index, ending and mid index of arry.

5. Use loop to traverse the array on the condition true which is starting is not greater then ending also element is not equal to mid of arry.
6. If key is greater than mid then change starting= mide+1.
7. If key is less than mid then change end= mide-1.
8. Find new mid using modified value of start and end then check again.
9. After termination of loop start>end means not found else found.
10. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code

```
void binarysearch(int arr[], int size, int key)
{
    int s=0;
    int e=size-1;
    int mid=(s+e)/2;

    while (s<=e && key!=arr[mid])
    {
        if(key>arr[mid])
        {
            s=mid+1;
        }
        else if(key<arr[mid])
        {
           e=mid-1;
        }

        mid=(s+e)/2;
    }

    if(s>e)
    {
       cout<<"Not found"<<endl;
    }
    else
    {
        cout<<"Found"<<endl;
    }

}// End of Void binary Search
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Semester\DSA\My_cod
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
Press 6 to search array through LinearSearch
Press 7 to search array through BinarySearch
Press 0 to exit from code
1
----------Initializing array------------
2 33 44 9
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 7 to search array through BinarySearch
Press 0 to exit from code
7
...........Binary Searching..........
Enter key/element
44
Found
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 7 to search array through BinarySearch
Press 0 to exit from code
7
...........Binary Searching..........
Enter key/element
66
Not found
Press 1 to initialize array
Press 6 to search array through LinearSearch
Press 7 to search array through BinarySearch
Press 0 to exit from code
```

Description of void Selection sort

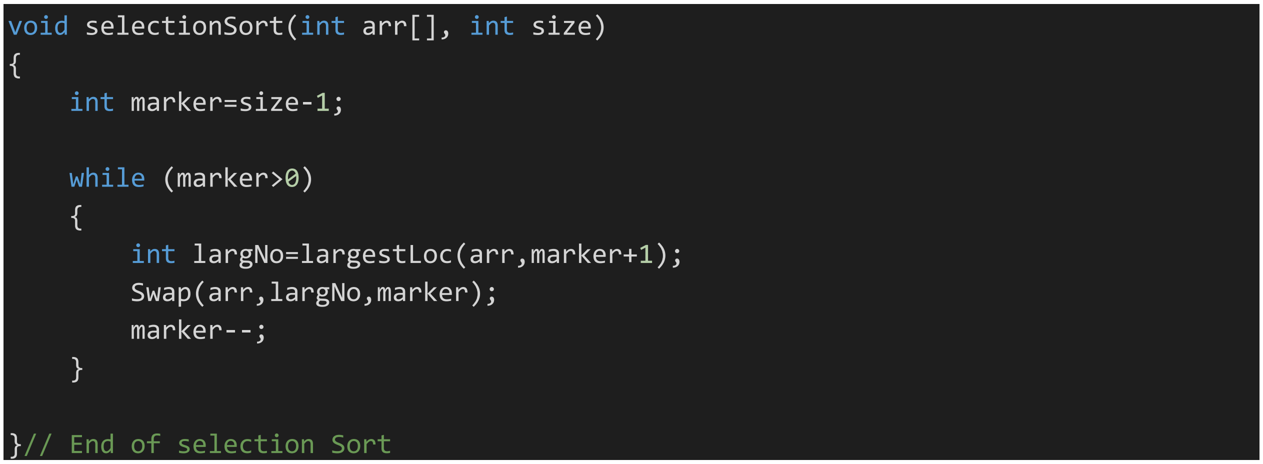
```
/*write a Function name selection sort  that will an arry,
size of arry  as parametrs, this function sort the arry in Acending order*/
void selectionSort(int arr[], int size);// Section sort
```

What does Function Do?

1. void function means **nothing will return.**
2. Take an array and size of array as a parameter.
3. Sort elements of array using selection sort method.
4. Sort in ascending order.
5. We initialized a variable marker which use for loop termination.
6. Use loop to traverse the array

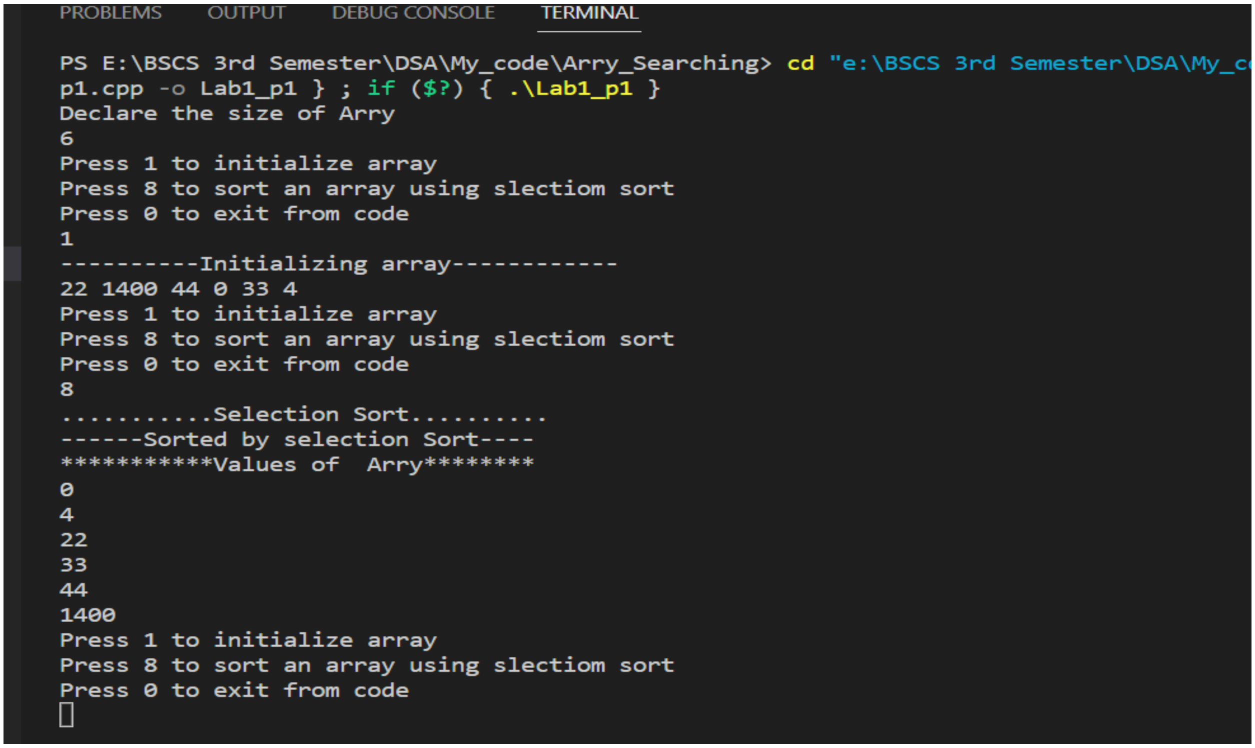
7. Find largest value location then swap with marker which makes largest value on marker then decrement the marker and perform again until marker not equal to zero.
8. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code



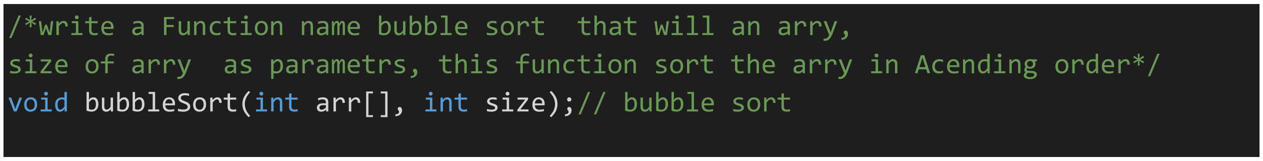
```
void selectionSort(int arr[], int size)
{
    int marker=size-1;

    while (marker>0)
    {
        int largNo=largestLoc(arr,marker+1);
        Swap(arr,largNo,marker);
        marker--;
    }

}// End of selection Sort
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Semester\DSA\My_c
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
6
Press 1 to initialize array
Press 8 to sort an array using slectiom sort
Press 0 to exit from code
1
----------Initializing array------------
22 1400 44 0 33 4
Press 1 to initialize array
Press 8 to sort an array using slectiom sort
Press 0 to exit from code
8
...........Selection Sort..........
------Sorted by selection Sort----
***********Values of  Arry********
0
4
22
33
44
1400
Press 1 to initialize array
Press 8 to sort an array using slectiom sort
Press 0 to exit from code
```

Description of void Bubble sort

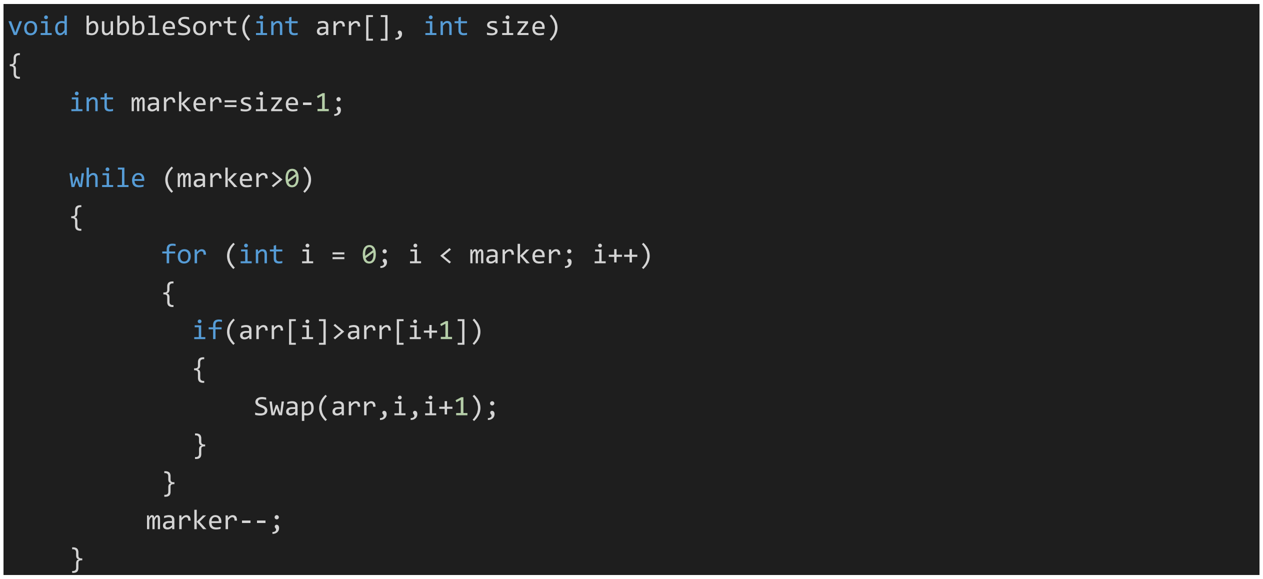


```
/*write a Function name bubble sort  that will an arry,
size of arry  as parametrs, this function sort the arry in Acending order*/
void bubbleSort(int arr[], int size);// bubble sort
```

What does Function Do?

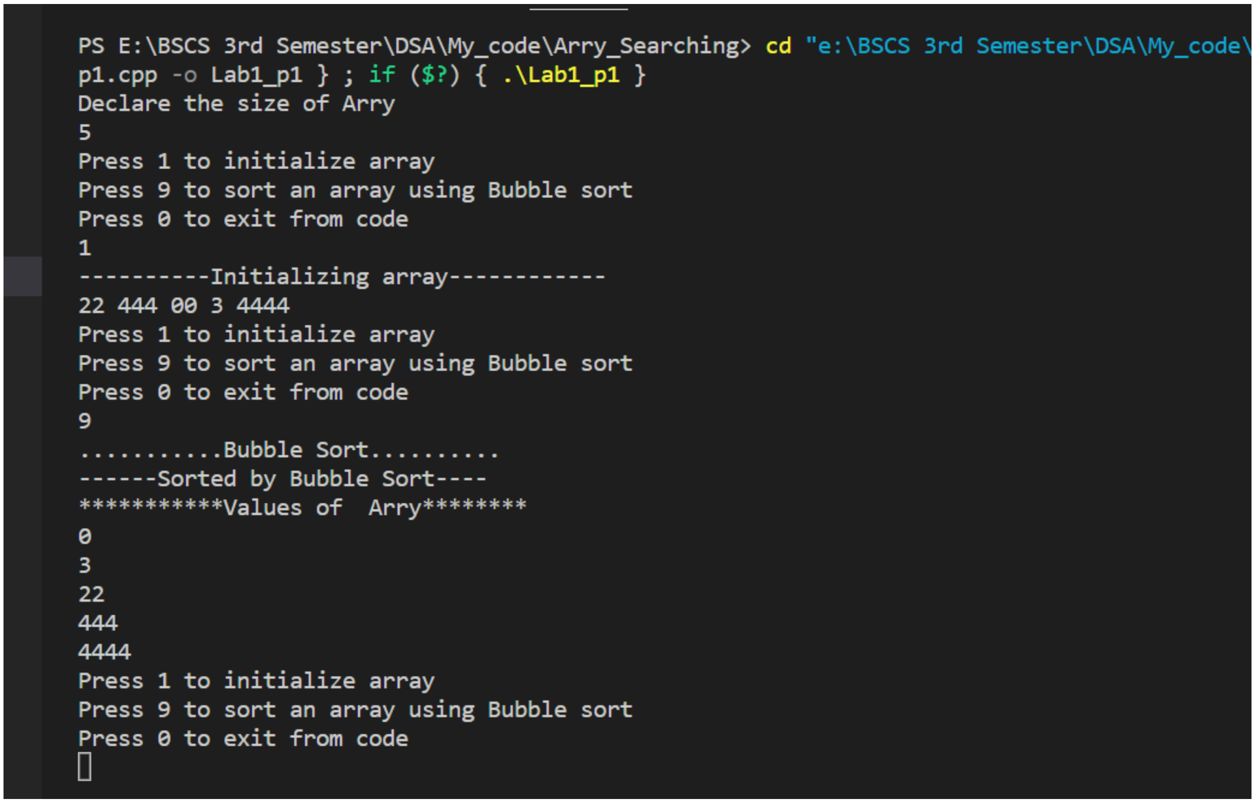
1. void function means **nothing will return.**
2. Take an array and size of array as a parameter.
3. Sort elements of array using bubble sort method.
4. Sort in ascending order.
5. We initialized a variable marker which use for loop termination.
6. Use loop to traverse the array
7. Within loop we compare the index of array with his next if index of array is greater than his next then swap (indexi, index i+1) and marker--;
8. It will iterate until marker is not equal to zero.
9. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code



```
void bubbleSort(int arr[], int size)
{
    int marker=size-1;

    while (marker>0)
    {
          for (int i = 0; i < marker; i++)
          {
            if(arr[i]>arr[i+1])
            {
                Swap(arr,i,i+1);
            }
          }
         marker--;
    }
}// End of Bubble sort
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Semester\DSA\My_code\
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
5
Press 1 to initialize array
Press 9 to sort an array using Bubble sort
Press 0 to exit from code
1
----------Initializing array-----------
22 444 00 3 4444
Press 1 to initialize array
Press 9 to sort an array using Bubble sort
Press 0 to exit from code
9
...........Bubble Sort..........
------Sorted by Bubble Sort----
***********Values of  Arry********
0
3
22
444
4444
Press 1 to initialize array
Press 9 to sort an array using Bubble sort
Press 0 to exit from code
```

Description of void insertion sort

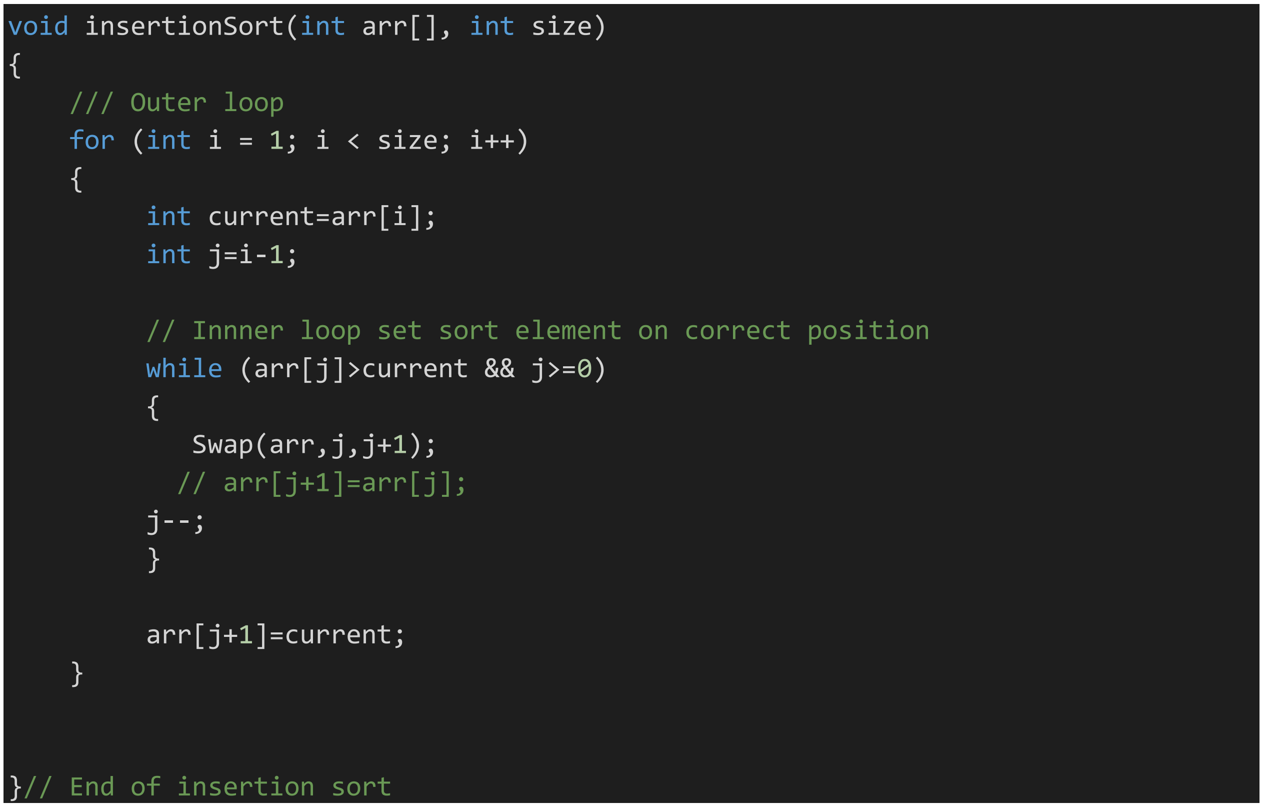
```
/*write a Function name bubble sort  that will an arry,
size of arry  as parametrs, this function sort the arry in Acending order*/
void insertionSort(int arr[], int size);// insertion sort
```

What does Function Do?

1. Void function means **nothing will return.**

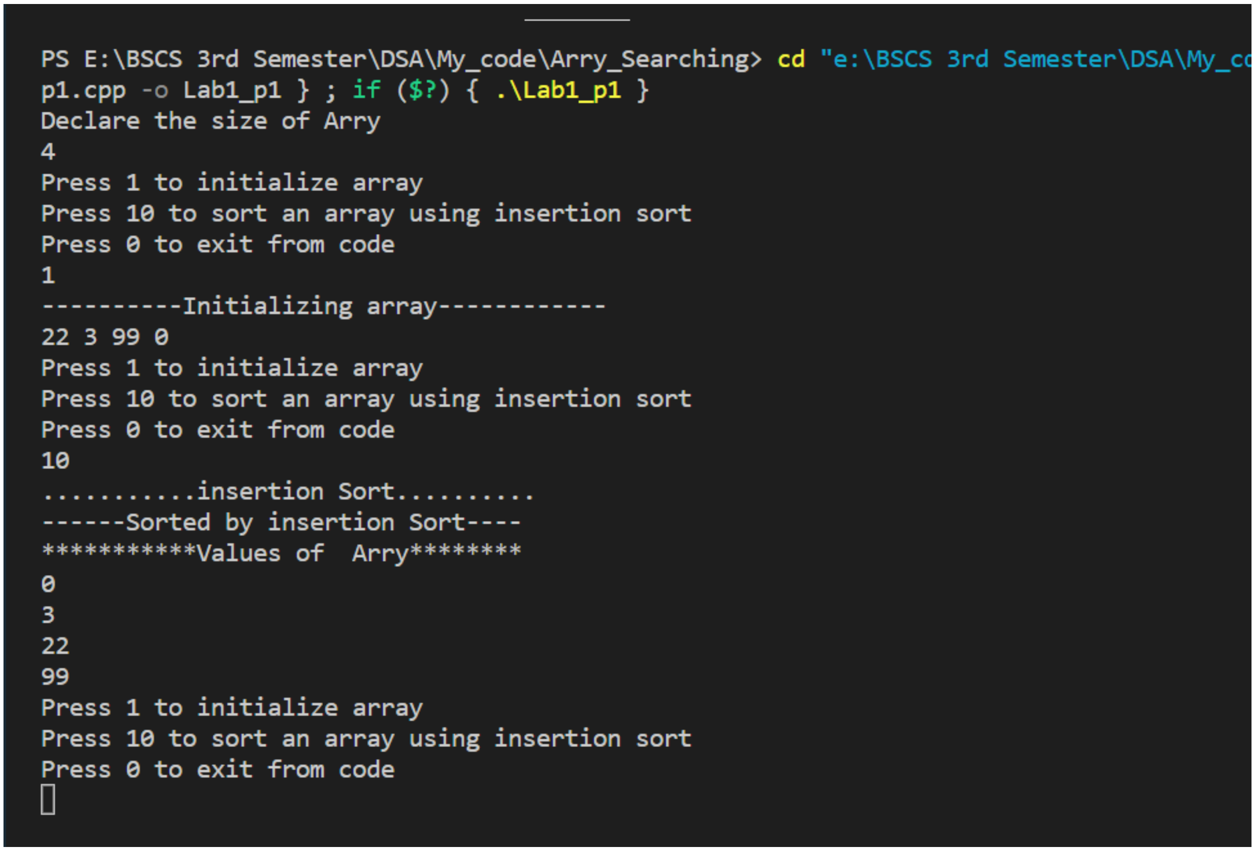
2. Take an array and size of array as a parameter.
3. Sort elements of array using insertion sort.
4. In insertion sort we compare the first element to all other element of array if element is greater than any other element then swap index. And we will perform this for all element of array.
5. We need nested loop.
6. Outer loop to traverse the array on the condition less than size of array.
7. In outer loop define the current (which store the index that will compare by his previous index j=i-1;) equal to arr[i].
8. Inner loop compares the element with other elements of array.
9. Condition of inner loop element is greater than current or index is not less than zero.
10. If condition satisfied swap (element to his next element) then decrement index
11. Use **switch** for menu display which helps to perform task according to **choice of user.**

Source Code



```
void insertionSort(int arr[], int size)
{
    /// Outer loop
    for (int i = 1; i < size; i++)
    {
         int current=arr[i];
         int j=i-1;

         // Innner loop set sort element on correct position
         while (arr[j]>current && j>=0)
         {
            Swap(arr,j,j+1);
           // arr[j+1]=arr[j];
         j--;
         }

         arr[j+1]=current;
    }


}// End of insertion sort
```

Output



```
PS E:\BSCS 3rd Semester\DSA\My_code\Arry_Searching> cd "e:\BSCS 3rd Semester\DSA\My_c
p1.cpp -o Lab1_p1 } ; if ($?) { .\Lab1_p1 }
Declare the size of Arry
4
Press 1 to initialize array
Press 10 to sort an array using insertion sort
Press 0 to exit from code
1
----------Initializing array------------
22 3 99 0
Press 1 to initialize array
Press 10 to sort an array using insertion sort
Press 0 to exit from code
10
...........insertion Sort..........
------Sorted by insertion Sort----
***********Values of  Arry********
0
3
22
99
Press 1 to initialize array
Press 10 to sort an array using insertion sort
Press 0 to exit from code
```

End of Lab 02