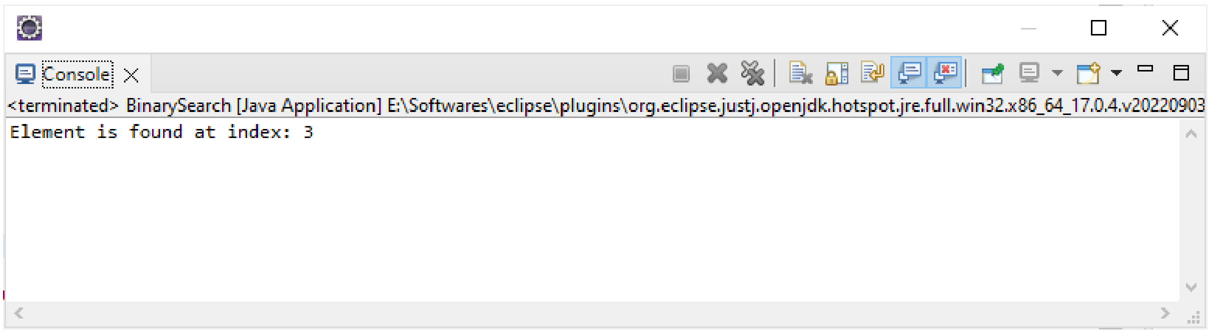
a. Perform Binary search operation

```
package com.binary.info;

public class BinarySearch{
 public static void binarySearch(int arr[], int first, int last, int
key){
   int mid = (first + last)/2;
   while( first <= last ){
      if ( arr[mid] < key ){
        first = mid + 1;
      }else if ( arr[mid] == key ){
        System.out.println("Element is found at index: " + mid);
        break;
      }else{
         last = mid - 1;
      }
      mid = (first + last)/2;
   }
   if ( first > last ){
      System.out.println("Element is not found!");
   }
 }
 public static void main(String args[]){
        int arr[] = {11,22,33,44,55};
        int key = 44;
        int last=arr.length-1;
        binarySearch(arr,0,last,key);
 }
}
```

Output:

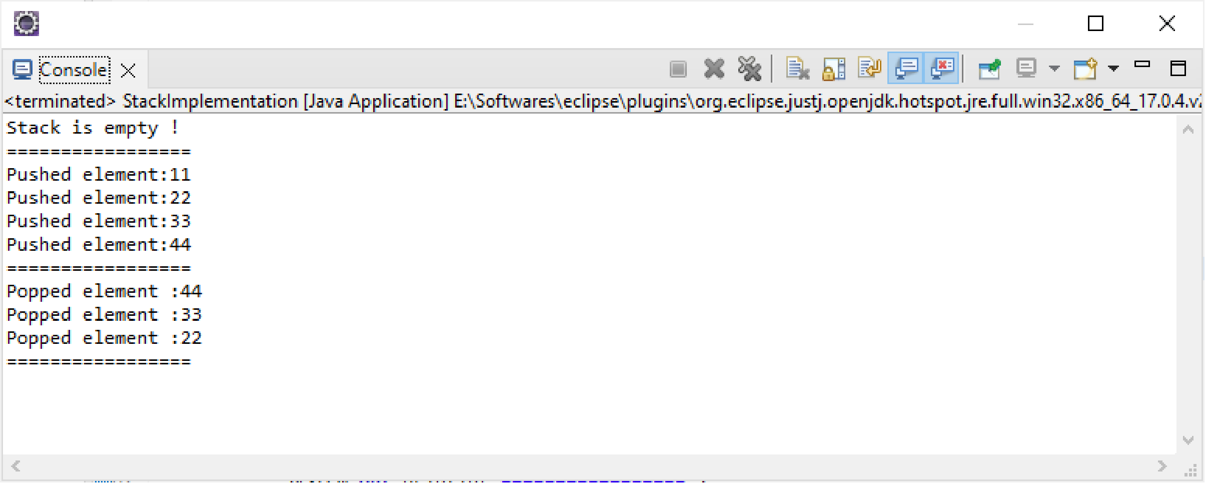


```
<terminated> BinarySearch [Java Application] E:\Softwares\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220903
Element is found at index: 3
```

b. Implement stack using array concepts

```
package com.implementStack.info;

     public class StackImplementation {
         int size;
         int arr[];
         int top;

         StackImplementation(int size) {
             this.size = size;
             this.arr = new int[size];
             this.top = -1;
         }

         public void push(int pushedElement) {
             if (!isFull()) {
                 top++;
                 arr[top] = pushedElement;
                 System.out.println("Pushed element:" + 
pushedElement);
             } else {
                 System.out.println("Stack is full !");
             }
         }

         public int pop() {
             if (!isEmpty()) {
                 int returnedTop = top;
                 top--;
                 System.out.println("Popped element :" + 
arr[returnedTop]);
                 return arr[returnedTop];

             } else {
                 System.out.println("Stack is empty !");
                 return -1;
             }
         }

         public int peek() {
             if(!this.isEmpty())
                             return arr[top];
                     else
                     {
                             System.out.println("Stack is 
Empty");
                             return -1;
                     }
         }

         public boolean isEmpty() {
             return (top == -1);
         }

         public boolean isFull() {
             return (size - 1 == top);
         }

         public static void main(String[] args) {
             StackImplementation StackCustom = new 
StackImplementation(10);
             StackCustom.pop();
             System.out.println("=================");
             StackCustom.push(11);
             StackCustom.push(22);
             StackCustom.push(33);
             StackCustom.push(44);
             System.out.println("=================");
             StackCustom.pop();
             StackCustom.pop();
             StackCustom.pop();
             System.out.println("=================");
         }
     }
```

Output:



```
<terminated> StackImplementation [Java Application] E:\Softwares\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v
Stack is empty !
=================
Pushed element:11
Pushed element:22
Pushed element:33
Pushed element:44
=================
Popped element :44
Popped element :33
Popped element :22
=================
```