

1. Write a Java program to
 - a. Perform Binary search operation

```
package com.dsa.q1;

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[] = { 10, 20, 30, 40, 50 };
        int key = 30;
        int last = arr.length - 1;
        binarySearch(arr, 0, last, key);
    }
}
```

```
Console ×
<terminated> BinarySearch [Java Application] C:\Users\hp\.p2\pool\pl
Element is found at index: 2
```

b. Implement stack using array concepts

```
package com.dsa.q2;
public class StackCustom {
    int size;
    int arr[];
    int top;

    StackCustom(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) {
        StackCustom StackCustom = new StackCustom(10);
        StackCustom.pop();
        System.out.println("=====");
        StackCustom.push(10);
        StackCustom.push(30);
        StackCustom.push(50);
        StackCustom.push(40);
        System.out.println("=====");
        StackCustom.pop();
        StackCustom.pop();
        StackCustom.pop();
        System.out.println("=====");
    }

```

}

```
Console ×
<terminated> StackCustom [Java Application] C:\Us
Stack is empty !
=====
Pushed element:10
Pushed element:30
Pushed element:50
Pushed element:40
=====
Popped element :40
Popped element :50
Popped element :30
=====
```