

LAB-7

Name: V.SAIKRISHNA

Reg.No.:19BCE7638

1Q.Binary Search.Java Code.

```
import java.io.*;
import java.util.*;

public class Main {
static int binarySearch(int[] a, int x)
{
int left = 0, right = a.length - 1;
    while (left <= right) {
        int mid = left + (right - left) /
        2;

        if (x == a[mid]) {
            return mid;
        } else if (x <
            a[mid]) { right =
                mid - 1;
        } else {
            left = mid + 1;
        }
    }
    return -1;
}
```

```
static int linearSearch(int[] a, int x)
{
    for (int i = 0; i < a.length; i++) {
        if (a[i] == x)
            return i;
    }
    return -1;
}
```

```
public static void main(String[] args) {
    FastScanner scanner = new
    FastScanner(System.in); int n =
    scanner.nextInt();
    int[] a = new int[n];
    for (int i = 0; i < n; i++) {
        a[i] =
        scanner.nextInt();
    }
    int m =
    scanner.nextInt(); int[]
    b = new int[m];
    for (int i = 0; i < m; i++)
    {
        b[i] = scanner.nextInt();
    }
}
```

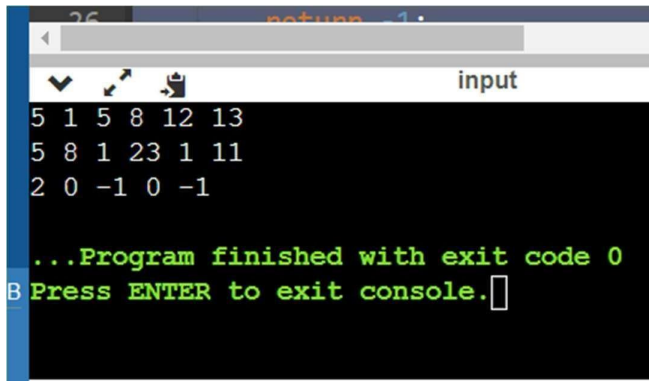
```
for (int i = 0; i < m; i++) {  
  
    System.out.print(binarySearch(a, b[i]) + " ");  
  
}  
}
```

```
static class FastScanner {  
    BufferedReader br;  
    StringTokenizer st;  
  
    FastScanner(InputStream  
        stream) { try {  
        br = new BufferedReader(new  
            InputStreamReader(stream));  
    } catch (Exception  
        e) {  
        e.printStackTrace  
        ();  
    }  
}
```

```
String next() {  
    while (st == null ||  
        !st.hasMoreTokens()) { try {  
        st = new StringTokenizer(br.readLine());  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    }  
    return st.nextToken();  
}
```

```
int nextInt() {  
    return Integer.parseInt(next());  
}  
}  
}
```

Output:

A screenshot of a Java IDE's console window. The window has a title bar with a file name '26' and a return value '-1'. Below the title bar is a toolbar with icons for running, debugging, and other IDE functions. The main area of the console is black with white text. It shows three lines of input: '5 1 5 8 12 13', '5 8 1 23 1 11', and '2 0 -1 0 -1'. Below these, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor at the end.

```
26 return -1
input
5 1 5 8 12 13
5 8 1 23 1 11
2 0 -1 0 -1
...Program finished with exit code 0
Press ENTER to exit console.
```

2Q: Max votes.

Code:

```
import
```

```
java.util.*;
```

```
import java.io.*;
```

```
public class Main {
```

```
    private static int getMaxVote(int[] a, int left, int right)
```

```
    { if (left == right) {
```

```
        return -1;
```

```
    }
```

```

    if (left + 1 == right) {
        return a[left];
    }
    int left_elem = getMaxVote(a, left, (left + right - 1) / 2 +
1); int right_elem = getMaxVote(a, (left + right - 1) / 2 +
1, right);

    int lcount = 0;
    for (int i = left; i < right; i++)
        { if (a[i] == left_elem)
            lcount += 1;
        }
    if (lcount > (right - left) / 2)
        return left_elem;

    int rcount = 0;
    for (int i = left; i < right; i++)
        { if (a[i] == right_elem)
            rcount += 1;
        }
    if (rcount > (right - left) / 2)
        return right_elem;
    return -1;
}

```

```

public static void main(String[] args) {
    FastScanner scanner = new
    FastScanner(System.in); int n =
    scanner.nextInt();
    int[] a = new int[n];
    for (int i = 0; i < n; i++) {
        a[i] =
        scanner.nextInt();
    }
    if (getMaxVote(a, 0, a.length) !=
        -1) { System.out.println(1);
    } else {
        System.out.println(0);

    }
}

```

```

static class FastScanner {
    BufferedReader br;
    StringTokenizer st;

    FastScanner(InputStream
    stream) { try {
        br = new BufferedReader(new
        InputStreamReader(stream));
    } catch (Exception e) {

```

```

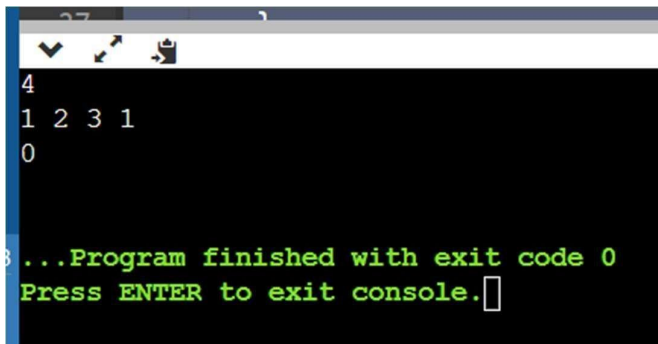
        e.printStackTrace();
    }
}

String next() {
    while (st == null ||
           !st.hasMoreTokens()) { try {
        st = new StringTokenizer(br.readLine());
    } catch (IOException e) {
        e.printStackTrace();
    }
    }
    return st.nextToken();
}

int nextInt() {
    return Integer.parseInt(next());
}
}

```

Output:

A screenshot of a terminal window with a black background and green text. The output shows the number '4' on the first line, followed by '1 2 3 1' on the second line, and '0' on the third line. Below these, a message states '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor at the end of the second line.

```
4
1 2 3 1
0
...Program finished with exit code 0
Press ENTER to exit console.
```

2Q. Maximum

Votes import

java.util.*; import

java.io.*;

public class Main {

private static int getMaxVote(int[] a, int left, int right)

{ if (left == right) {

return -1;

}

if (left + 1 == right)

{ return a[left];

}

int left_elem = getMaxVote(a, left, (left + right - 1) / 2 +
1); int right_elem = getMaxVote(a, (left + right - 1) / 2 +
1, right);

int lcount = 0;

for (int i = left; i < right;

i++) { if (a[i] ==

left_elem)

lcount += 1;

}

if (lcount > (right - left) / 2)

return left_elem;

int rcount = 0;

```
for (int i = left; i < right;
    i++) { if (a[i] ==
    right_elem)
        rcount += 1;
}

if (rcount > (right - left) / 2)
```

```

        return
        right_elem; return
        -1;
    }

```

```

public static void main(String[] args) {

    FastScanner scanner = new
    FastScanner(System.in); int n =
    scanner.nextInt();
    int[] a = new int[n];
    for (int i = 0; i < n; i++) {
        a[i] =
        scanner.nextInt();
    }

    if (getMaxVote(a, 0, a.length) != -1) {
        System.out.println(1);
    } else {

        System.out.println(0);

    }

}

```

```

static class
    FastScanner {
        BufferedReader br;
        StringTokenizer st;

        FastScanner(InputStream
        stream) { try {
            br = new BufferedReader(new
            InputStreamReader(stream));

```

```
    } catch (Exception  
        e) {  
        e.printStackTrace  
        ();  
    }  
}
```

```
String next() {  
    while (st == null || !st.hasMoreTokens()) {  
        try {
```

```
        st = new StringTokenizer(br.readLine());  
    } catch (IOException  
        e) {  
        e.printStackTrace();  
    }  
}  
return st.nextToken();  
  
}  
  
int nextInt() {  
    return Integer.parseInt(next());  
}  
  
}
```



The screenshot shows a console window titled "input" with a black background and white text. The output of the program is as follows:

```
5  
2 3 9 2 2  
1  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```