1) Write a program to print the sum of all the elements present on even indices in the given array?

Ans:

```
public class EvenIndexSum {
    public static void main(String[] args) {
        int[] arr = { 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 };
        int sum = 0;

        for (int i = 0; i < arr.length; i += 2) {
            sum += arr[i];
        }

        System.out.println("Sum of elements at even indices: " + sum);
    }
}
```

2) Write a program to traverse over the elements of the array using for each loop and print all even elements.

Ans:

```
public class EvenElementTraversal {
    public static void main(String[] args) {
        int[] arr = { 10, 21, 30, 43, 50, 62, 75, 80, 95, 100 };

        System.out.println("Even elements in the array:");

        for (int element : arr) {
            if (element % 2 == 0) {
                System.out.println(element);
            }
        }
    }
}
```

3) write a program to calculate the maximum elements in a given array.

Ans:

```
public class MaxElementInArray {
    public static void main(String[] args) {
        int[] arr = { 10, 25, 7, 42, 68, 30, 91, 53, 77, 12 };

        int maxElement = arr[0]; // Assume the first element as the maximum

        for (int i = 1; i < arr.length; i++) {
            if (arr[i] > maxElement) {
                maxElement = arr[i];
            }
        }

        System.out.println("Maximum element in the array: " + maxElement);
    }
}
```

4) Write a program to find out the second largest element in a given array.

Ans:

```
public class SecondLargestElement {
    public static void main(String[] args) {
        int[] arr = { 10, 25, 7, 42, 68, 30, 91, 53, 77, 12 };

        int largest = Integer.MIN_VALUE; // Initialize with smallest possible value
        int secondLargest = Integer.MIN_VALUE; // Initialize with smallest possible value

        for (int num : arr) {
            if (num > largest) {
                secondLargest = largest;
                largest = num;
            } else if (num > secondLargest && num != largest) {
                secondLargest = num;
            }
        }

        if (secondLargest != Integer.MIN_VALUE) {
            System.out.println("Second largest element: " + secondLargest);
        } else {
            System.out.println("There is no second largest element.");
        }
    }
}
```

5) Given an array. Find the first peak element in the array. A peak element is an element greater than its just left and just right neighbor.

Ans:

```
public class FirstPeakElement {
    public static void main(String[] args) {
        int[] arr = { 5, 10, 20, 15, 7, 25, 30 };

        int peakIndex = findPeakElementIndex(arr);

        if (peakIndex != -1) {
            System.out.println("The first peak element is: " + arr[peakIndex]);
        } else {
            System.out.println("There is no peak element in the array.");
        }
    }

    public static int findPeakElementIndex(int[] arr) {
        int n = arr.length;

        if (n == 0) {
            return -1; // No peak in an empty array
        }

        if (n == 1 || arr[0] > arr[1]) {
            return 0; // First element is a peak if it's greater than the second
        }

        if (arr[n - 1] > arr[n - 2]) {
            return n - 1; // Last element is a peak if it's greater than the second last
        }

        // Check for peaks in the middle
        for (int i = 1; i < n - 1; i++) {
            if (arr[i] > arr[i - 1] && arr[i] > arr[i + 1]) {
                return i; // Current element is a peak
            }
        }

        return -1; // No peak found
    }
}
```