2: Find maximum sum path involving elements of given arrays Given two sorted arrays of integers, find a maximum sum path involving elements of both arrays whose sum is maximum.

For example,

Input: X = { 3, 6, 7, 8, 10, 12, 15, 18, 100 } Y = {1, 2, 3, 5, 7, 9, 10, 11, 15, 16, 18, 25, 50}

The maximum sum path is: 1 -> 2 >3->6-7-9->10-> 12 > 15->16 -> 18 -> 100

The maximum sum is 199

********CODE*************

```
public class MaximumSumPath {
    public static void main(String[] args) {
        int[] X = {3, 6, 7, 8, 10, 12, 15, 18, 100};
        int[] Y = {1, 2, 3, 5, 7, 9, 10, 11, 15, 16, 18, 25, 50};

        int sumX = 0, sumY = 0, maxSum = 0;
        int indxX = 0, indxY = 0;

        while (indxX < X.length && indxY < Y.length) {
            if (X[indxX] < Y[indxY]) {
                sumX =  sumX+ X[indxX++];
            }
            else if (X[indxX] > Y[indxY]) {
                sumY += Y[indxY++];
            }
```

```
          else {
              // When elements are equal, take the maximum sum till now and add current common
element
              maxSum = maxSum+ Math.max(sumX, sumY) + X[indxX];
              sumX = 0;
              sumY = 0;
              indxX++;
              indxY++;
          }
      }

      // Add remaining elements of X and Y
      while (indxX < X.length) {
          sumX = sumX + X[indxX++];
      }

      while (indxY < Y.length) {
          sumY = sumY+ Y[indxY++];
      }

      // Add the maximum sum of remaining elements
      maxSum = maxSum + Math.max(sumX, sumY);

      System.out.println("Maximum sum : " + maxSum);
  }
}
```

```
//OUTPUT //

Maximum sum : 199
```

3:Q3:Write a Java Program to count themumber of words in a string using HashMap

```
import java.util.HashMap;

public class WordCount {
    public static void main(String[] args) {
        String inputString = "Hello Everyone ! Hello guys Good morning everyone";
        String[] words = inputString.trim().split("\\s+");

        // Create a HashMap to store word counts
        HashMap<String, Integer> wordCountMap = new HashMap<>();
         for (String word : words) {
            String lowercaseWord = word.toLowerCase();

            wordCountMap.put(lowercaseWord, wordCountMap.getOrDefault(lowercaseWord, 0) + 1);
        }
         System.out.println("Word count in the string:");
        for (String word : wordCountMap.keySet()) {
            System.out.println(word + ": " + wordCountMap.get(word));
        }
    }
}
```

```
/** OUTPUT ***/
Word count in the string:
!: 1
everyone: 2
guys: 1
hello: 2
good: 1
morning: 1
string.: 1
world!: 1
sample: 1
hello: 1
```

Q4:Write a java program to find the duplicate characters in a string

```
public class FindDUPLICATEcharacter {

        public static void main(String[] args) {
                String s1 ="abcdabcaba";
String s2="";

for(int i=0;i<s1.length();i++)

{

            int cnt=0;

            for(int j=i+1;j<s1.length();j++)

            {



                        if(s1.charAt(i)== s1.charAt(j))

                                    cnt++;

            }

            if(cnt>0)

                        s2=s2+s1.charAt(i);



}

   System.out.println("DUPLICATE CHARACTER: "+s2);

            }



}
```

```
***********OUTPUT**********

DUPLICATE CHARACTER: abcaba
```