**/\* A- method accepts given strings and return 2 valus :**

**1- longest string**

**2 - number of chars of longest string \*/**

**/\* public class Main {**

**public static void main(String[] args) {**

**String[] strings = {"Java", "Python", "JavaScript", "C++"};**

**StringPair longestStringPair = findLongestString(strings);**

**System.out.println("Longest string: " + longestStringPair.getString());**

**System.out.println("Number of characters: " + longestStringPair.getLength());**

**}**

**public static StringPair findLongestString(String[] strings) {**

**String longestString = "";**

**for (String str : strings) {**

**if (str.length() > longestString.length()) {**

**longestString = str;**

**}**

**}**

**return new StringPair(longestString, longestString.length());**

**}**

**}**

**class StringPair {**

**private String string;**

**private int length;**

**public StringPair(String string, int length) {**

**this.string = string;**

**this.length = length;**

**}**

**public String getString() {**

**return string;**

**}**

**public int getLength() {**

**return length;**

**}**

**} \*/**

**// B- method accepts given strings and diacritic marks in Czech and return string without diacritic marks in Czech //**

**/\* import java.text.Normalizer;**

**public class Main {**

**public static void main(String[] args) {**

**String[] strings = {"Český", "řeřicha", "účast"};**

**String[] processedStrings = removeDiacriticMarks(strings);**

**for (int i = 0; i < strings.length; i++) {**

**System.out.println("Original: " + strings[i]);**

**System.out.println("Processed: " + processedStrings[i]);**

**System.out.println();**

**}**

**}**

**public static String[] removeDiacriticMarks(String[] strings) {**

**String[] processedStrings = new String[strings.length];**

**for (int i = 0; i < strings.length; i++) {**

**String processedString = Normalizer.normalize(strings[i], Normalizer.Form.NFD);**

**processedString = processedString.replaceAll("\\p{InCombiningDiacriticalMarks}+", "");**

**processedStrings[i] = processedString;**

**}**

**return processedStrings;**

**}**

**}**

**\*/**

**/\***

**C-Write a function that takes in a string of one or more words, and returns the same string,**

**but with all five or more letter words reversed. Strings passed in will consist of only letters and spaces.**

**Spaces will be included only when more than one word is present.**

**\*/**

**public class Main {**

**public static void main(String[] args) {**

**String input = "Hello, world! This is a test";**

**String reversedWords = reverseLongWords(input);**

**System.out.println("Reversed words: " + reversedWords);**

**}**

**public static String reverseLongWords(String input) {**

**String[] words = input.split(" ");**

**StringBuilder result = new StringBuilder();**

**for (String word : words) {**

**if (word.length() >= 5) {**

**StringBuilder reversedWord = new StringBuilder(word).reverse();**

**result.append(reversedWord).append(" ");**

**} else {**

**result.append(word).append(" ");**

**}**

**}**

**return result.toString().trim();**

**}**

**}**