|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task1 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("N: "); |
|  | int n = Integer.parseInt(input.nextLine()); |
|  | System.out.printf("K: "); |
|  | int k = Integer.parseInt(input.nextLine()); |
|  | System.out.printf("String: "); |
|  | String text = input.nextLine(); |
|  | System.out.println(essay(n, k, text)); |
|  | } |
|  |  |
|  |  |
|  | public static String essay(int n, int k, String text) { |
|  | String[] words = text.split(" "); |
|  | String currentString = ""; |
|  | String result = ""; |
|  | boolean added = false; |
|  | if (n > words.length) return "can't create essay with this parameters"; |
|  | for (int i = 0; i < words.length; i++) { |
|  | String word = words[i]; |
|  | if (currentString.replace(" ", "").length() + word.length() <= k) currentString += word + " "; |
|  | else { |
|  | result += currentString + "\n"; |
|  | currentString = word + " "; |
|  | } |
|  | } |
|  | if (currentString.length() > 0) { |
|  | result += currentString + "\n"; |
|  | } |
|  | return result; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task2 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("String: "); |
|  | String text = input.nextLine(); |
|  | System.out.println(balance(text)); |
|  | } |
|  |  |
|  |  |
|  | public static List<String> balance(String text) { |
|  | List<String> result = new ArrayList<String>(); |
|  | int balance = 0; |
|  | String current = ""; |
|  | for (char ch: text.toCharArray()) { |
|  | if (Character.toString(ch).equals("(")) { |
|  | balance++; |
|  | current += "("; |
|  | } |
|  | else { |
|  | balance--; |
|  | current += ")"; |
|  | } |
|  | if (balance == 0) { |
|  | result.add(current); |
|  | current = ""; |
|  | } |
|  | } |
|  | return result; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task3 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("String: "); |
|  | String text = input.nextLine(); |
|  | if (text.indexOf("\_") != -1) { |
|  | System.out.printf("toCamelCase: %s\n", toCamelCase(text)); |
|  | } |
|  | else { |
|  | System.out.printf("toSnakeCase: %s\n", toSnakeCase(text)); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | public static String toCamelCase(String text) { |
|  | char[] chars = text.toCharArray(); |
|  | String result = ""; |
|  | for (int i = 0; i < chars.length; i++) { |
|  | if (String.valueOf(chars[i]).equals("\_")) { |
|  | chars[i+1] = Character.toUpperCase(chars[i + 1]); |
|  | continue; |
|  | } |
|  | result += String.valueOf(chars[i]); |
|  | } |
|  | return result; |
|  | } |
|  |  |
|  |  |
|  | public static String toSnakeCase(String text) { |
|  | char[] chars = text.toCharArray(); |
|  | String result = ""; |
|  | for (int i = 0; i < chars.length; i++) { |
|  | if (Character.toUpperCase(chars[i]) == chars[i]) { |
|  | result += "\_"; |
|  | chars[i] = Character.toLowerCase(chars[i]); |
|  | } |
|  | result += String.valueOf(chars[i]); |
|  | } |
|  | return result; |
|  |  |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task4 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("Start: "); |
|  | double start = Double.parseDouble(input.nextLine()); |
|  | System.out.printf("End: "); |
|  | double end = Double.parseDouble(input.nextLine()); |
|  | System.out.printf("Salary: "); |
|  | double salary = Double.parseDouble(input.nextLine()); |
|  | System.out.printf("Salary for overtime: "); |
|  | double overtimeSalary = Double.parseDouble(input.nextLine()); |
|  | System.out.println(overtime(start, end, salary, overtimeSalary)); |
|  | } |
|  |  |
|  |  |
|  | public static String overtime(double start, double end, double salary, double overtimeSalary) { |
|  | String money = "$"; |
|  | if (end > 17) return money + String.valueOf((17 - start) \* salary + (end - 17) \* salary \* overtimeSalary); |
|  | return money + String.valueOf((end-start) \* salary); |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task5 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("Mass: "); |
|  | String m = input.nextLine(); |
|  | System.out.printf("Height: "); |
|  | String h = input.nextLine(); |
|  | System.out.println(BMI(m, h)); |
|  | } |
|  |  |
|  |  |
|  | public static String BMI(String m, String h) { |
|  | double mass = 0; |
|  | double height = 0; |
|  | Map<String, Double> metrics = new HashMap<String, Double>(); |
|  | metrics.put("pounds", 2.205); |
|  | metrics.put("kilos", 1.0); |
|  | metrics.put("inches", 39.37); |
|  | metrics.put("meters", 1.0); |
|  | for (Map.Entry<String,Double> entry: metrics.entrySet()) { |
|  | if (m.indexOf(entry.getKey()) != -1) { |
|  | mass = Double.parseDouble(m.replace(entry.getKey(), "").trim()) / entry.getValue(); |
|  | } |
|  | if (h.indexOf(entry.getKey()) != -1) { |
|  | height = Double.parseDouble(h.replace(entry.getKey(), "").trim()) / entry.getValue(); |
|  | } |
|  | } |
|  | double res = mass / Math.pow(height, 2); |
|  | if (res < 18.5) return String.format("%.1f", res) + " Underweight"; |
|  | else if (res >= 18.5 && res < 24.9) return String.format("%.1f", res) + " Normal weight"; |
|  | return String.format("%.1f", res) + " Overweight"; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task6 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("Number: "); |
|  | Integer number = Integer.parseInt(input.nextLine()); |
|  | System.out.println(bugger(number)); |
|  | } |
|  |  |
|  |  |
|  | public static int bugger(int n) { |
|  | String value = String.valueOf(n); |
|  | int count = 0; |
|  | int result = 1; |
|  | while (value.length() != 1) { |
|  | for (int i = 0; i < value.length(); i++) {; |
|  | result \*= Integer.parseInt(String.valueOf(value.charAt(i))); |
|  | } |
|  | count++; |
|  | value = String.valueOf(result); |
|  | result = 1; |
|  | } |
|  | return count; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task7 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("Input: "); |
|  | String string = input.nextLine(); |
|  | System.out.println(toStarShorthand(string)); |
|  | } |
|  |  |
|  |  |
|  | public static String toStarShorthand(String input) { |
|  | String result = ""; |
|  | int count = 1; |
|  | char currentChar = new Character('a'); |
|  | for (int i = 1; i < input.length(); i++) { |
|  | currentChar = input.charAt(i - 1); |
|  | if (currentChar == input.charAt(i)) count++; |
|  | else { |
|  | if (count > 1) result += currentChar + "\*" + count; |
|  | else result += currentChar; |
|  | count = 1; |
|  | if (i == input.length() - 1) result += String.valueOf(input.charAt(input.length() - 1)); |
|  | } |
|  | } |
|  | if (count != 1) result += currentChar + "\*" + count; |
|  | return result; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  | import java.util.regex.Matcher; |
|  | import java.util.regex.Pattern; |
|  |  |
|  | public class Task8 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("First string: "); |
|  | String first = input.nextLine(); |
|  | System.out.printf("Second string: "); |
|  | String second = input.nextLine(); |
|  | System.out.println(doesRhyme(first, second)); |
|  | } |
|  |  |
|  |  |
|  | public static boolean doesRhyme(String f, String s) { |
|  | String[] words1 = f.split(" "); |
|  | String word1 = words1[words1.length - 1]; |
|  | String[] words2 = s.split(" "); |
|  | String word2 = words2[words2.length - 1]; |
|  | String vowels = "EeAaUuIiOoYy"; |
|  | List<String> first = new ArrayList<String>(); |
|  | List<String> second = new ArrayList<String>(); |
|  | boolean equal = true; |
|  | for (char ch: vowels.toCharArray()) { |
|  | String c = String.valueOf(ch); |
|  | if (word1.indexOf(c) != -1) { |
|  | first.add(c.toLowerCase()); |
|  | } |
|  | if (word2.indexOf(c) != -1) { |
|  | second.add(c.toLowerCase()); |
|  | } |
|  | } |
|  | if (first.size() != second.size()) return false; |
|  | for (int i = 0; i < first.size(); i++) { |
|  | if (equal) { |
|  | equal = false; |
|  | for (int j = 0; j < second.size(); j++) { |
|  | if (second.get(j).equals(first.get(i))) equal = true; |
|  | } |
|  | } |
|  | } |
|  | System.out.printf("First: %s, Second: %s\n", first, second); |
|  | return equal; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task9 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("First number: "); |
|  | String num1 = input.nextLine(); |
|  | System.out.printf("Second number: "); |
|  | String num2 = input.nextLine(); |
|  | System.out.println(trouble(num1, num2)); |
|  | } |
|  |  |
|  |  |
|  | public static boolean trouble(String number1, String number2) { |
|  | int count1 = 0; |
|  | int count2 = 0; |
|  | String currentChar = ""; |
|  | for (int i = 0; i < number1.length() - 1; i++) { |
|  | currentChar = String.valueOf(number1.charAt(i)); |
|  | if (currentChar.equals(String.valueOf(number1.charAt(i + 1)))) count1++; |
|  | else count1 = 0; |
|  | if (count1 == 2) { |
|  | count2 = 0; |
|  | for (int j = 0; j < number2.length(); j++) { |
|  | if (currentChar.equals(String.valueOf(number2.charAt(j)))) count2++; |
|  | else count2 = 0; |
|  | if (count2 == 2) return true; |
|  | } |
|  | } |
|  | } |
|  | return false; |
|  | } |
|  | } |

|  |
| --- |
| import java.util.\*; |
|  |  |
|  |  |
|  | public class Task10 { |
|  | public static void main(String[] args) { |
|  | Scanner input = new Scanner(System.in); |
|  | System.out.printf("String: "); |
|  | String string = input.nextLine(); |
|  | System.out.printf("Book end: "); |
|  | Character bookend = input.nextLine().charAt(0); |
|  | System.out.println(unique(string, bookend)); |
|  | } |
|  |  |
|  |  |
|  | public static int unique(String book, Character end) { |
|  | //String[] books = book.split(end); |
|  | Map<Character, Boolean> chars = new HashMap<Character, Boolean>(); |
|  | boolean parse = false; |
|  | for (char c: book.toCharArray()) { |
|  | if (parse && c != end) chars.put(c, true); |
|  | if (c == end) { |
|  | if (parse) parse = false; |
|  | else parse = true; |
|  | } |
|  | } |
|  | System.out.println(chars); |
|  | return chars.size(); |
|  | } |
|  | } |