

1. Write a Java program to trim any leading or trailing whitespace from a given string
2. We'll say that a "mirror" section in an array is a group of contiguous elements such that somewhere in the array, the same group appears in reverse order. For example, the largest mirror section in {1, 2, 3, 8, 9, 3, 2, 1} is length 3 (the {1, 2, 3} part). Return the size of the largest mirror section found in the given array.

maxMirror([1, 2, 3, 8, 9, 3, 2, 1]) → 3

maxMirror([1, 2, 1, 4]) → 3

maxMirror([7, 1, 2, 9, 7, 2, 1]) → 2

3. Given a string and a non-empty **word** string, return a string made of each char just before and just after every appearance of the word in the string. Ignore cases where there is no char before or after the word, and a char may be included twice if it is between two words.

wordEnds("abcXY123XYijk", "XY") → "c13i"

wordEnds("XY123XY", "XY") → "13"

wordEnds("XY1XY", "XY") → "11"

4. Given arrays nums1 and nums2 of the same length, for every element in nums1, consider the corresponding element in nums2 (at the same index). Return the count of the number of times that the two elements differ by 2 or less, but are not equal.

matchUp([1, 2, 3], [2, 3, 10]) → 2

matchUp([1, 2, 3], [2, 3, 5]) → 3

matchUp([1, 2, 3], [2, 3, 3]) → 2

5. Write a program to check if the word 'orange' is present in the "This is orange juice".

6. Write a program to delete all consonants from the string "Hello, have a good day"

Input a string of alphabets. Find out the number of occurrence of all alphabets in that string. Find out the alphabet with maximum occurrence.