**My Hamming Dna**

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under submit directory and specific file names for each exercise are also provided at the beginning of each exercise under submit file(s).

| **My Hamming Dna** |  |
| --- | --- |
| Submit directory | ex00 |
| Submit file | my\_hamming\_dna\* |
| Languages | javascript => .js, python => .py, ruby => .rb, c => .c, ... |

**Description**

Calculate the Hamming difference between two DNA strands.

A mutation is simply a mistake that occurs during the creation or copying of a nucleic acid, in particular DNA. Because nucleic acids are vital to cellular functions, mutations tend to cause a ripple effect throughout the cell. Although mutations are technically mistakes, a very rare mutation may equip the cell with a beneficial attribute. In fact, the macro effects of evolution are attributable by the accumulated result of beneficial microscopic mutations over many generations.

The simplest and most common type of nucleic acid mutation is a point mutation, which replaces one base with another at a single nucleotide.

By counting the number of differences between two homologous DNA strands taken from different genomes with a common ancestor, we get a measure of the minimum number of point mutations that could have occurred on the evolutionary path between the two strands.

This is called the 'Hamming distance'.

It is found by comparing two DNA strands and counting how many of the nucleotides are different from their equivalent in the other string.

GAGCCTACTAACGGGAT

CATCGTAATGACGGCCT

^ ^ ^ ^ ^ ^^

The Hamming distance between these two DNA strands is 7.

Your function will return an integer

Given 2 strings with the exact number of characters, count how many characters are different at the same position. if they don't share the exact number of characters, return -1.

if the two arguments are not the same size, you will return -1

**Example 00**

Input: "GGACTGA" && "GGACTGA"

Output:

Return Value: 0

**Example 01**

Input: "ACCAGGG" && "ACTATGG"

Output:

Return Value: 2

**Example 02**

Input: "GGACGGATTCTG" && "AGG"

Output:

Return Value: -1

**Example 03**

Input: "" && ""

Output:

Return Value: 0