



[BACK](#)

Eye Rhyme

 159,454



[<](#) [DESCRIPTION](#) [MY SOLUTIONS](#) [LEADERBOARD](#) [COMMENTS](#) [README](#) [>](#)

RECOVERY SCORE: 100/100

An *eye rhyme* is a rhyme in which two words are spelled similarly but pronounced differently. An example is the pair `cough` and `bough` ; although they look similar, when they are spoken there is no rhyming quality.

You are writing a thesis on the *eye rhyme*, and you thought it would be cool to make the text itself *eye rhymed*. This brilliant idea came to your mind a little too late: the text is already written. Now you want to check if a given pair of lines in your text have an *eye rhyme*. More specifically, you want to make sure that the last three characters of each pair of lines coincide.

You have already split your text into pairs of lines. Now all that's left is to check that the last three characters of the lines in each `pairOfLines` coincide. Implement a function that will do this job.

Example

- For `pairOfLines = "cough\tbough"` , the output should be `eyeRhyme(pairOfLines) = true` .  
  
Both lines end with `ugh` .
- For `pairOfLines = "CodeFig!ht\tWith all your might"` , the output should be `eyeRhyme(pairOfLines) = false` .  
  
The first line ends with `!ht` , and the second one ends with `ght` .

Input/Output

- [execution time limit] 4 seconds (js)
- [input] string `pairOfLines`  
  
A string in the format `"<line1>\t<line2>"` , where `<linei>` consists of at least 3 characters and may contain any character except `'\t'` (tabulation character). The lines are separated by `'\t'` (tabulation character).  
  
Guaranteed constraints:  
`7 ≤ pairOfLines.length ≤ 1000` .
- [output] boolean  
  
`true` if the lines in `pairOfLines` have an *eye rhyme*, `false` otherwise.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
}
```