* The challenge asks to find the top 10 most repeated words in the matrix.
* It also says the search results should only count the repeated words in the stream once. By that I assume that the program should discard repeated words in the given stream
* I also assume I should try to clean the stream before finding words in the matrix, discarding the words with a length higher than the Nº of rows AND columns.
* Given the challenge asks to find the top 10 most repeated words in the matrix, I think the algorithm should return a list of the words from the wordStream that are inside the matrix, but order them by the Nº of times each word appeared in the matrix (descending)
* I have 2 ideas to solve this:
  1. Take each word in the given stream and try to find it in the matrix by checking every letter in the matrix and check to the right, left, above & below that letter if the word appears
     + With this idea I should only take each word from the wordStream (and maybe its reverse) and check almost all the characters in the matrix (if looking for the word and its reverse I will never reach the matrix letters near the left and bottom edges).  
       Since the matrix’s max size is 64x64, at most, the time it takes to process should be something like:  
        **O(64\*64\*N) --> N = nº words in stream**  
         
       **NOTE** --> The above is not entirely accurate, cause I have to consider the other loops I make inside the algorithm
  2. After doing the above algorithm and trying to find a better solution I found about “Trie” structures, which are useful to represent words in a tree structure, and made a Depth-First Search algorithm

For all the ideas I have to consider the following:

* Instead of searching in every direction, I could reverse the words in the stream and check if, for each letter in the matrix, either the word or its reverse appears in the matrix (that way I should only search in 2 directions: down and right, if starting from the [0, 0] coordinate in the matrix)