

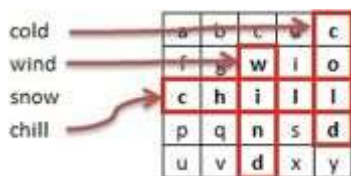


## Developer Challenge: Word Finder

**Objective:** The objective of this challenge is not necessarily just to solve the problem - but to evaluate your software development skills, code quality, analysis, creativity, and resourcefulness as a potential future colleague. Please share the necessary artifacts you would provide to your colleagues in a real-world professional setting to best evaluate your work.

---

Presented with a character matrix and a large stream of words, your task is to create a Class that searches the matrix to look for the words from the word stream. Words may appear horizontally, from left to right, or vertically, from top to bottom. In the example below, the word stream has four words and the matrix contains only three of those words ("chill", "cold" and "wind"):



The search code must be implemented as a class with the following interface:

```
public class WordFinder
{
    public WordFinder(IEnumerable<string> matrix) {
        ...
    }
    public IEnumerable<string> Find(IEnumerable<string> wordstream)
    {
        ...
    }
}
```

The WordFinder constructor receives a set of strings which represents a character matrix. The matrix size needs to be 64x64, all strings contain the same number of characters.

The "Find" method should return the top 10 most repeated words from the word stream found in the matrix. If no words are found, the "Find" method should return an empty set of strings. If any word in the word stream is found more than once within the matrix, the search results should count it only once.

Due to the size of the word stream, the code should be implemented in a **high performance** fashion both in terms of efficient algorithm and utilization of system resources. Where possible, please include your analysis and evaluation.

**We should run this application pressing F5 and/or executing dotnet run**