Hi,

Please send an email reply to confirm you've received this description.

Goal

Write a program in C to find all words from a word list file (such as /usr/share/dict/words or /usr/dict/words) that can be found in a 4x4 grid of letters.

The grid should be given as input to the program. Specify the grid as one 16-character argument or 16 1-character arguments to the program or in some other reasonable way.

If you don't have a words file, contact me at once and I can email you a copy. Create a local git repository, and regularly commit your changes so that we can see your progress.

As soon as you have a working, debugged program, commit any remaining outstanding changes and then email the git repository containing all source code to me.

To do this, execute these commands from within your repository:

```
git bundle create <yourname>.git --all
tar cvfj <yourname>.tar.bz2 <yourname>.git
```

We ask you to do this so we can tell about how much time it took. Please do not spend more than 3 hours. Any submissions beyond 3 hours will not be considered, so please submit your code within the allotted time, regardless of how far you've progressed.

Do not send large attachments. The specific word list you used is not important, so do not send it.

Problem Details

A word can be found in a grid by starting on any letter, then moving to an adjacent letter and so on.

Example 1.

```
a b a n
z q z d
r r o r
r n r r
```

The word "abandon" fits in the grid first horizontally, then diagonally down and to the left. The word "ran" does NOT fit in the grid because 'r' and 'a' are not adjacent.

A word cannot use a specific letter in the grid more than once.

Example 2.

The grid

```
z z z z
z z b z
h z e z
a z z z
```

contains the words "be", "ha", and "ah". It should not contain "bee" or "ebb" or "he" or "hah".

Implementation Details

Your program should not print any word more than once. Your program can assume that the word list file contains no duplicated words and that the input arguments are valid.

Your program should be tolerant of word files that contain a mixture of upper and lower case words and of words that contain punctuation. Your program can deal with such words in some reasonable way. Note that different word list files can be alphabetized differently.

Please do not copy or examine any of your or other people's existing code while writing this program. Please write it from scratch. (Exception: you may use the file I/O code you wrote as part of the prerequisites mentioned previously.) We realise your program will just be a prototype. We do not expect this to be production quality software.

Work quickly, but accurately because we take into consideration how fast you're able to write the program. However, make sure your program works correctly.

Strive for simplicity in your code and data structures. A simple, concise program is more impressive than a long, complicated one and also easier to write quickly.

Don't spend a significant amount of time writing comments.

If it's easy to do, please specify the grid on the command line as the last argument or arguments and print the results to the standard output. An example program run might be:

```
a.out /usr/share/dict/words abanzqzdrrorrnrr
...
abandon
...
```

Your output should be a list of matching words separated by newlines. Don't print out anything else unless your program is reporting an error condition.

Good luck!