

CPS 109 Computer Science I, Assignment 2

Objective: To practice using 2D arrays, loops and comparisons.

General Application: a crossword puzzle game

Source: Horstmann, Big Java

Instructions

Arrays store a linear arrangement of values, accessed by a single index. Two-dimensional arrays or *matrices* store a tabular arrangement of values, accessed by two indices, for example `matrix[i][j]`, where `i` is the row index, and `j` is the column index.

Crosswords are a type of puzzle where the "pieces" of the puzzle have letters in common. For example, these pieces share the letter 'o'.

```
      d
a c r o s s
      w
      n
```

Write a program that will accept a list of words into an array list of strings, then make a 20 x 20 matrix of characters (strings of length 1) that contains a crossword puzzle with these words.

For example, when given the list
addle, apple, clowning, incline, plan, burr
your program might display:

```
  a d d l e
    p
    p
c l o w n i n g
  e         n
           c
           l
           i
    p l a n
           e
```

Below is an example which is **not** legal because of words touching each other. Note how "prove" is added in a fashion which is illegal since it creates additional words "dr", "do", "lv", and "ee". Similarly, the addition of "no" vertically is illegal since it creates the extra word "no" horizontally. You do not have to check a dictionary in this assignment -- just assume that creation of any new word other than the one you are adding is illegal.

```
  a d d l e
  p r o v e
    p
c l o w n i n g
  e         n o
           c
           l
           i
    p l a n
           e
```

Here is another example of how a word can be added legally -- the addition of "loon" uses two letters that are in the puzzle already, which creates no problem since there are no new words created other than the single word added "loon".)

```

a d d l e
p      o
p      o
c l o w n i n g
e          n
          c
          l
          i
p l a n e

```

To summarize: A **legal** placement of a word *W* has the following properties:

- i. All of the word *W* lies on the 20x20 grid;
- ii. If *W* is not the first word, then *W* must intersect with one or more words on the grid;
- iii. The placement of *W* must not create any new 'words', i.e., *W* cannot be adjacent to any letter which is not in the word(s) with which *W* intersects.

Hint (optional approach):

1. Place the first word horizontally in the center of the grid.
2. Place the next word vertically (so that it intersects the first word), and keep alternating horizontal with vertical placement.
3. When placing a word *W*, scan the current grid to find a letter of *W*, and check if *W* can legally be placed to overlap with that letter. If yes, then place *W* in that position on the grid. If no, then continue looking through the grid for some letter of *W* to try another placement.
4. If it is not possible to place *W* anywhere, then skip that word, and try the next word. When you skip a word, print the word and a message: "**no matching letter**" or "**illegal adjacencies**" or "**reaches outside grid**".
5. Continue in this fashion until all words have been either placed or at least tried and reported on.
6. When done going through the list of words and placing them if possible, print the crossword puzzle.

Notes:

1. Write your program in a file called **a2.py**, which contains
 - a) a function **crossword(L)**, which makes and prints a crossword puzzle for the words in list *L*, and
 - b) code that runs **crossword(L)** on at least two lists of words.
2. Submit two files on D2L: **a2.py** and **output.txt**, where **output.txt** contains the results from your tests in 1(b) above.

3. There are multiple ways to build a crossword puzzle from the given list of words -- many different outputs can be correct for the same list of words. However, if a word *can* be added to the crossword at the time your program reads it, then your program must add it.
4. **Marking: 9 points** for producing correct output; **1 point** for documentation and readability.
5. Put your **name** in a comment at the top of the file. If you receive significant help from someone, then acknowledge that person's help in the comment at the top. This is part of the documentation. Copying or letting your work be copied results in a mark of zero as well as other repercussions according to the student code of conduct described in the Ryerson Calendar.