Assignment 3(bonus): WordBrain

1 Introduction

1.1 Due Date

This assignment is due May 2 2021 at 11:59:00 (one minute before noon)

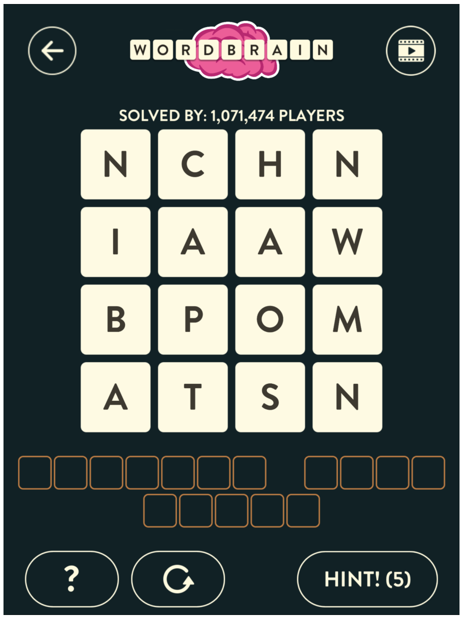
1.2 Group Size

For this assignment, the maximum group size is 3

1.3 About wordbrain

The smartphone app wordbrain is a puzzle game. The game presents users with a grid of letters, ranging in size 2x2 up to 5x5 and even larger.

Here is an example:



N	C	H	N
I	A	A	W
B	P	O	M
A	T	S	N

The user is given the pattern of word lengths. For this puzzle, the first word to be solved for is 7 letters, then the second word is 4 letters, and the final word is 5 letters.

The words must be spelled with adjacent letters, and as correct words are formed, the remaining letters drop down.

The game play is easier to understand if you have the app, so please download it and try it.

2 The assignment: wordbrainsolver

Your program will be given the filenames of two word lists, via the command-line, like this:

```
./wordbrainsolver firstwordlist secondwordlist
```

The *firstwordlist* is a file containing a small word list, and the *secondwordlist* is a file containing a more comprehensive word list.

Wordlists should be represented in trie data structure.

The reason for two word lists is that the puzzle answer is more likely to use basic words than exotic / rare words, so the result of the small word list will be much faster, and if it finds an answer, usually the correct answer.

Your program will read from standard input a description of a puzzle to solve, in the following format:

```
nchn
iaaw
bpom
atsn
**** ***** *** ****
```

The first lines represent the word grid, and the line with asterisks represent the words to solved for. [Any letters specified in this last line are already known.]

Your program will then print out all possible solutions to the puzzle using the first word list as the set of possible words.

If there are none, then all possible solutions using the second word list will be printed.

2.1 More specifications

- The program should output a period '.' on a new line at the conclusion of each puzzle solution, whether the puzzle had a solution or not.
- The output must be printed in sorted order. For this puzzle, using the provided word lists, the following should be printed:

```
snow panic man bath
.
```

- The user of your program can continue to enter puzzles to be solved, without limit.
- There is no limit to the puzzle size: the puzzle might be 1x1 or 10x10 or 331x331
- The program only exits when EOF (ctrl-D) is encountered, or the user enters an empty line at the beginning of a puzzle.
- The puzzles entered will all be in valid formats.

Your program must be called wordbrainsolver.cpp

2.2 Example

The result of running

```
./wordbrainsolver small_word_list.txt large_word_list.txt <puzzles.txt >solutions.txt
```

when *puzzles.txt* is

```
hee
oqr
sua
*** ******
yson
elnn
hnca
olab
***** ***** ******
nchn
iaaw
bpom
atsn
**** ***** *** ****
vanmo
ipveo
toarr
tsmed
miipb
**** ******* ******* *******
vanmo
ipveo
toarr
tsmed
miipb
p*** ******* ******* *******
yson
elnn
hnca
```

```
olab
***** holly ******
```

is that the output is redirected to the file solutions.txt and that output should be

```
hoe square
.
banes holly cannon
hones bally cannon
honey balls cannon
.
snow panic man bath
.
opts bedroom vampire vitamin
post bedroom vampire vitamin
pots bedroom vampire vitamin
stop bedroom vampire vitamin
.
post bedroom vampire vitamin
pots bedroom vampire vitamin
.
banes holly cannon
.
```

The program should exit on an empty line of input or EOF.

2.3 Extra words

The following words have been added to both word lists as they appear frequently in the game:

- tv
- keyring

2.4 Downloads

- small_word_list.txt
- large_word_list.txt
- puzzles.txt
- solutions.txt