Data Structures and Objects
CSIS 3700

Spring Semester 2019 — CRN 21212

Project 3 — Word Ladders
Due date: Friday, April 12, 2019

Goal

Develop and implement a word ladder finder.

Details

A word ladder (*aka* doublet) is a type of puzzle created by Charles Dodgson (*aka* Lewis Carroll) in the 1800s. The premise of the puzzle is to "convert" one word into the other by a sequence of single-letter changes, with each change generating a valid English word. For example, you can change *beer* into *wine* via the sequence *beer – beet – bent – lent – lint – line – wine*. There may be other ladders for these words as well.

Your program should begin by reading words from the file **`sgb-words.txt`**; the file consists of 5 757 words used to test the Stanford GraphBase. Note that all words have exactly five letters.

Each word should have two values connected to it:

- A pointer to a word (and its connected data)
  Pro tip: You can use an integer index instead of a pointer. Use `−1` for **`NULL`** or **`nullptr`**.
- A linear list of pointers to other words

Initialize the pointer for each word to **`NULL`**. Then, examine each pair of words — there are 16 568 646 such pairs. If the pair has a *Hamming distance* of 1, then add each word — a pointer to the word, actually — to the other word's list.

Read two five-letter words from the keyboard. Find both words in the word list. If either word is not in the list, stop and output that no word ladder exists. Otherwise, do the following algorithm.

Algorithm 1 Generating a word ladder

Preconditions w_1 and w_2 are words in the word list

Postconditions S contains a word ladder from w_1 to w_2

```
procedure GenLadder($w_1$,$w_2$)
    Clear $S$

    Add $w_2$ to a queue $Q$

    while $Q$ is not empty do
        Dequeue $Q$ into $w$

        for each word $v$ in $w.list$ do
            if $v.ptr$ = NULL and $v \neq w_2$ then
                $v.ptr \leftarrow w$
                Enqueue $v$ in $Q$
            end if
        end for
    end while

    if $w_1.ptr \neq$ NULL then
        Append $w_1$ to $S$
        $w \leftarrow w_1.ptr$

        while $w \neq$ NULL do
            Append $w$ to $S$
            $w \leftarrow w.ptr$
        end while
    end if
end procedure
```

After running the algorithm, if S is empty then there is no ladder. Otherwise, display the contents of S in order to obtain the ladder.

What to turn in

Turn in your source code and `Makefile`. If you are using an IDE, compress the folder containing the project and submit that.

Example 1

▹*Input*

```
while loops
```

▹*Output*

```
Ladder:
while
whine
chine
chins
coins
loins
loons
loops
```

Example 2

▹*Input*

```
zzzzz yyyyy
```

▹*Output*

```
Error: first word not in list
```

Example 3

▹*Input*

```
books zzzzz
```

▹*Output*

```
Error: second word not in list
```

Example 4

▹*Input*

```
there their
```

▹*Output*

```
No ladder exists
```