CSE 2010, Term Project, Fall 2018
Due Wed Nov 28 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxInitial
Due Wed Dec 5 at the start of your lab section; Submit Server: class = cse2010, assignment = termProjectSxFinal
x is 2, 3, 4, or j—your section number (or java)

On a smartphone, the keyboard is small and typing is not as easy as on a regular-size keyboard. Wouldn't you like to be able to type easier and faster? How would you design a system that can help you and other users type quickly and accurately?

The challenge of the term project is to design and build SmartWord, a system that can guess words based on partial words. For each word, given a letter at a time, your system will provide 3 guesses. If one of the guesses is correct, the rest of the letters are skipped (yeah!). You and other users desire accuracy, speed and low memory usage, which can be measured by:

- accuracy (average % of letters skipped for each word)
- average time used to generate 3 guesses for a word
- number of bytes used in the memory
- $score = accuracy^2/\sqrt{time * memory}$

Note that an iPhone user can type 30 words per minute[1] and the average word length is 5.1 letters[2]. Hence, a user types on average 1 letter per 0.4 seconds. See how fast you can type at `http://10fastfingers.com/mobile`.

Program Files: Among your program files, you will proivde smartWord.c that supports at least:

- initSmartWord(...) // a collection of English words
- procOldMsgSmartWord(...) // user's old messages
- guessSmartWord(...) // 3 guesses on new messages
- feedbackSmartWord(...) // feedback for SmartWord

We will provide evalSmartWord.c (not to be modified) to help you evaluate your system. evalSmartWord.c and a template of smartWord.c with details of the above functions are on the course website.

You may use any program segment/file from the textbook. However, you may *NOT* use program segments/files from other sources. You may borrow ideas from other sources, but you need to clearly cite them at the top of your program files and implement them on your own.

Input: Input is from the command-line arguments for evalSmartWord.c in this order:

- filename for a list of known words, one word per line
- filename for old messages to simulate old user input, one message per line
- filename for new messages to simulate new user input, one message per line

Sample input files are on the course website.

You may have additional "hardcoded" (not from command line or prompt) input data files for smartWord.c and/or other program files.

Output: Measured performance by evalSmartWord.c goes to the standand output (screen).

Presentation: Project presentations (about 15 minutes each) will be during the lab on Dec 5 (Wed). Create a presentation file following this outline:

1. Title, Group Name, and Members
2. Goal and Motivation
3. Initial Approach/Submission
   (a) Algorithms and supporting data structures
   (b) Additional input data and reasons
   (c) Ideas devised within the group
   (d) Ideas discussed in the course/book, cite them
   (e) Ideas borrowed from other sources, cite them
4. Final Approach/Submission
   (a) Changes in algorithms and supporting data structures
   (b) Changes in additional input data and reasons
   (c) Ideas devised within the group
   (d) Ideas discussed in the course/book, cite them
   (e) Ideas borrowed from other sources, cite them
5. Evaluation: Accuracy, time, memory, and score: initial vs. final approach/submission for each data set
6. Analysis
   (a) Why more/less accurate?
   (b) Why faster/slower (including time complexity—n words, each word has m characters)?
   (c) Why more/less memory?
   (d) Possible further improvements

Evaluation:

- Initial submission (20%) – including performance on accuracy and score
- Final submission (40%) – including performance on accuracy and score
- Presentation (20%, oral and written)
- Teammate evaluation (20%)
- Extra Credit (a max of 10 points): Final Submission has an accuracy above 45, using less than 0.4 seconds and 2×10^8 bytes (~200 MB), as measured by evalSmartWord.c on data from multiple users. Average accuracy of at least 46 earns one extra point, at least 47 earns two extra points, ... Late submission is not applicable.

Submission: Initial submision has smartWord.c, other program files, and additional data files (if any).

Final submission has smartWord.c, other program files, additional data files (if any), and presentation (preferably in PDF).

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.

[1] `http://www.pocketables.com/archives/mobile-device-keyboard`
[2] `http://www.wolframalpha.com/input/?i=average+english+word+length`