

INTRODUCTION TO SCIENTIFIC AND ENGINEERING COMPUTATION

Assignment 4

May 2020

Objective: Find a hidden sentence in a book.

Description

An old way of secret communication is to construct sentences from the words in a book. The sender and receiver agree on a book and the sender provides the receiver with instructions for extracting the sentence from the book. Your program's job is to get a text file that contains the book and a text file that contains the instructions, and the hidden sentence by following the instructions, and print it.

Assignment

Your program that takes two command line arguments:

- The name of the file that contains the book. You can assume that there will be at most 10000 lines in the file, and that each line will be at most 200 characters long.
- The name of the file that contains the instructions. Each line in the file describes an instruction.

Each instruction specifies how to extract the next word. The first instruction gives the index of the line and the index of the word within the line. Note that both indexes are 1-based (not 0-based). Subsequent instructions specify the offset of the next line (positive offsets move forward, negative offsets move backward) and the index of the word in that line. For example if the instruction lines are 5 2, 2 4, -3 1, the words to be selected are: line 5 word 2, line 7 word 4, line 4 word 1.

Your program will read both files and store the data in two arrays. To represent the book, use a `char**` (an array of strings). To represent the instructions, use a struct `instruction_s*` where each instruction is represented by the following struct type:

```

struct instruction_s {
    int step; int index;
}

```

The step specifies the offset of the next line and the index specifies the index of the word in the line.

Write the following functions to accomplish your task:

- A function `get_word` that takes a line `s` and an index `n`, and returns the selected word. This function should have the following signature:

```
char* get_word(char* s, int n)
```

- A function that takes the lines of a book, a sequence of instructions, the number of instructions `n_instructions`, and an empty sentence where to store the extracted sentence. This function should have the signature:

```
void get_sentence(char** lines, struct instruction_s* instructions, int
n_instructions, char* sentence)
```

As you are processing the text, you have to remove all punctuation marks attached to the words (e.g. `hi!` should be converted to `hi`) and convert all upper case letters to lower case (e.g. `Hi!` should be converted to `hi`).

You can assume that the instructions will not result in locations that do not exist in the text.

Two files have been provided for testing your code: The `alice.txt` file contains the novel *Alice's Adventures in Wonderland* taken from Project Gutenberg, and the `instructions.txt` contains an example sequence of instructions. Below is an example run:

```
./assignment4 alice.txt instructions.txt
```

output:

```
fight for your right to party
```

Although not required in this assignment, you are strongly recommended to run your program under `valgrind` to spot any memory issues. Unfortunately `valgrind` is not installed in the system that we access using SSH. An example run using `valgrind` is as follows:

```
valgrind --leak-check=full ./assignment4 alice.txt instructions.txt
```

Rules

- Make sure to properly document your functions (purpose, parameters, etc.) as shown in the class and the slides.
- Your source code file has to have the name assignment4.c .
- Your program will be compiled using the following command on a Linux system. If it cannot be compiled and linked using this command, it will not be graded (failed submission).

```
gcc -std=c99 -Wall -Werror assignment4.c -o assignment4
```

- Your program will be checked using an automatic checker. Therefore, make sure you print the messages exactly as given in the example runs and check your code using given test file.
- You can use string library functions and file I/O functions.
- Do NOT use any C++ features such as cout and cin.
- Make sure your coding style is proper and consistent. Use the clang-format tool if necessary. Don't use any variable names in a language other than English.
- This is an individual assignment. Collaboration in any form is NOT allowed. No working together, no sharing code in any form including showing code to your classmates to give them ideas.
- All the code you submit must be your own. Don't copy/paste any piece of code from any resource including anything you've found on the Internet.
- The assignments will be checked for plagiarism using both automated tools and manual inspection. Any assignment involving plagiarism and/or infringement of intellectual property will be not be graded and is subject to further disciplinary actions.