Homework 4

100 Points

Structures and Strings

Project: Text Analyzer: Word Count (see next pages)

Grading

1. main() – 10
2. Get data from file – 40
3. Display words with highest frequency – 20
4. Write to file – 20
5. Report – 10

Write a short report (not more than one page) to explain the design of your program:

→ Show how data are organized in an array of structures
// What are the structures used in this program?
// What are the arrays used in this program?

→ Show the structure of your program
// structure chart or pseudocode
// What are the functions used in this program

Run the program as required and save the output at the end of the source file as a comment. Compress the source file, input and output files (if any), and the report, and upload the compressed file: 22B_LastName_FirstName_H4.zip

Project: Text Analyzer: Word Count (see next pages)

Write a program that parses a text file into words and counts the number of occurrences of each word. As you are reading the words, keep track of the number of words, maximum length, and highest frequency. Read a new word then search for it: if found, add one to the counter; if not found, insert the new word in the sorted list and initialize its counter to 1. Display the words with the highest frequency. Save the sorted list of words to a file as shown below:

```
6 words
Maximum length: 7
Highest frequency: 125

  9 a
 21 boat
125 dream
 44 merrily
125 row
  5 stream
```

You may assume that the text does not contain more than 100 words (i.e. declare an array of MAX_SIZE = 100, or use a vector) and that the frequency is not greater than 999 (for output formatting).

Prompt the user to enter the name of the input file. Run the program twice, using two input files: **`song_row.txt`** and **`song_ten.txt`** (see below). Assume the text files contain only letter, spaces and punctuation characters.

Create the input file **`song_row.txt`**, with the following data:

```
Row, row, row your boat,
Gently down the stream.
Merrily, merrily, merrily, merrily,
Life is but a dream.
```

Create the input file **`song_ten.txt`**, with the following data:

```
Ten green bottles hanging on the wall,
Ten green bottles hanging on the wall,
And if one green bottle should accidentally fall,
There will be nine green bottles hanging on the wall.
```

Extra Credit: Create and use an array of pointers to sort the words in descending order by frequency. Save the sorted array to another file, using the same format as described above.