資料結構 Project1

Word Frequency Counter

1. Description

Calculate the frequency of every word appering in the article (not including punctuation marks, space and special characters).
Output the k most frequently occuring words along with each frequency, the place of this word it appears at the first time , and execution time.

I. DO NOT use STL. Implement the entire program by yourself.
II. The algorithm is not limited but must be implemented with “array.”
III. Using C/C++
IV. Testing platform is codeblocks / dev-c++ / visual studio
V. Tip: You can use time.h ~~function~~ or unix command to calculate the execution time.
VI. Sorting is not required in this assignment. If required, you can use any kind of algorithm to sort. (Hint : It may not be faster to use sorting.)
VII. The “words ” are composed of English alphabet and numbers.

2. Input / Output

I. Input (File)

An article. Each word is separated by space, punctuation marks or new line.

II. Execution Process

File_name.txt
k

III. Output(Standard output)

Output k lines of the k most frequently occuring words along with each frequency in order. **Ex: (the,4,7).** Every pair is separated by new line. And output the execution time.

Attention:

i. Capital letter and small letter are the same.
Ex: “The” and “the” are the same word.

ii. The same frequency is considered the same rank. Must output the k most frequency occuring words.

Example 1

Input file:

A skunk sat on a stump. The skunk though the stump stunk; the stump thought the skunk stunk.

Execution:

```
input.txt
2
```

Output:

```
(the,4,7)
(skunk,3,2)
(stump,3,6)
0.002ms
```

Example 2

Input file:

I wish to wish the wish you wish to wish, but if you wish the wish the witch wishes, I won't wish the wish you wish to wish.

Execution:

```
Input.txt
3
```

Output:

```
(wish,11,2)
(the,4,5)
(to,3,3)
(you,3,7)
0.003ms
```

3. Testfile example

Download “test1_hw1.txt”from E3.
Note：This file is encoded with UTF-8. Please check out the encoding method if the punctuation is garbled.

4. Requirements

Program

I. You need to turn in the code, test file downloaded from e3 and executable file.

II. Name your executable file “StudenyID_hw1.” (Ex. 0116815_hw1)

III. Using standard output to print out your results and record them in your report.

IV. Your program must be readable (Ex. Comments, variable names, function names)

V. Please make sure the test file downloaded from e3 is executable.

Report (Name the file”StudentID_hw1.pdf”, **Ex: 0116815_hw1.pdf**)

I. Descript your implementation. (Ex: algorithm, program executing process)

II. Results: test file form E3, execution result and execution time.Please explain the results.

III. Others(Ex. Searching)

IV. No more than 2 pages.

5. Grading policy

I. Programing execution (60%)

II. Self-test(20%)

Run the testfile from e3 and record the result and execution time in your report.

III. Report (20%)

IV. Bonus (10%)

TA will supply a special testfile and you can get the bonus score if your result is cerrect.

6. Submit (e3 will be closed on time)

Compress all your files (including your code, test file downloaded from e3, executable file and report.) Name your compressed file “studentID_hw1.rar” or “studentID_hw1.zip”. Upload your compressed file to e3.

Deadline: 2013.11.19, 23:59

No late upload.

7. Attention

NO plagiarize.

If your file name is false, you will get zero.