Object-Oriented Programming Language

10/06/2016

Homework Assignment No. 3

Due 09:00 pm, Wednesday October 12, 2016

Late submission within 24 hours: score*0.9;

Late submission before post of solution: score*0.8 (the solution will usually be posted within a week); no late submission after the post of solution)

(Total 130%)

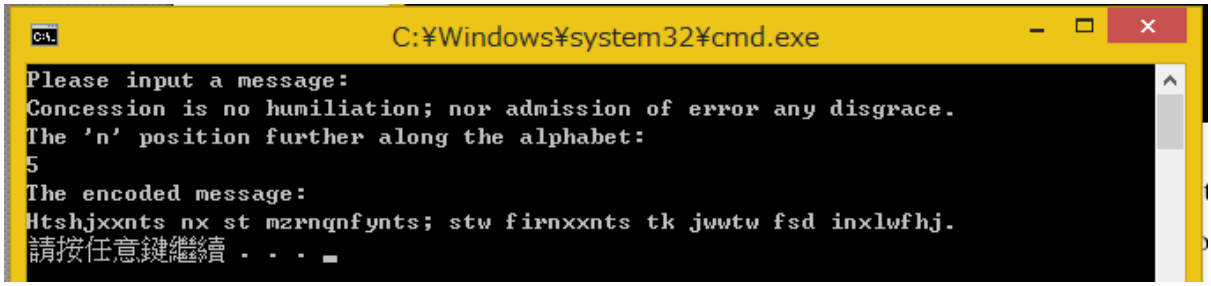
1. **(20%)** A textfile `input.txt` contains sentences of text. A line with an empty string indicates a paragraph break. Write a program to store all the lines in a vector container and print the lines in the first paragraph. For example, if our `input.txt` has the following contents (please copy and paste and generate the file by yourself):

```
Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim
Because it was grassy and wanted wear,
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I,
I took the one less traveled by,
And that has made all the difference.

The Road Not Taken by Robert Frost
```

A sample output looks like:

```
The first paragraph of the input file is

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;
```

2. **(30%)** A ***Caesar cipher*** encodes a message by replacing each letter of the message with the letter that is n positions further along in the alphabet. For example, if you have a letter `a`, its `2` positions in the alphabet is the letter `c`. Letters at the end of the alphabet wrap around to the beginning. For example, if you have a letter `y`, its `5` positions in the alphabet is the letter `d`. Write a program that takes a message to encode and an integer n as its parameters, and encodes the message using the n-shift. Below is a sample run:



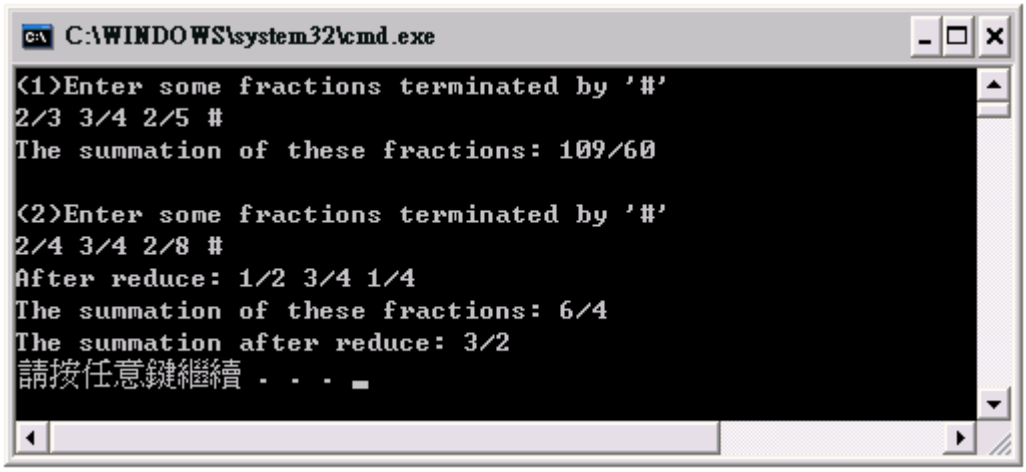
3. **(40%)** A fraction consisting of both a numerator and a denominator can be manipulated as a single object using the type `Fraction`:

```
struct Fraction{
    int num; //分子
    int den; //分母
};
```

Put the type definition in a header file `Fraction.h` with inclusive header guard.

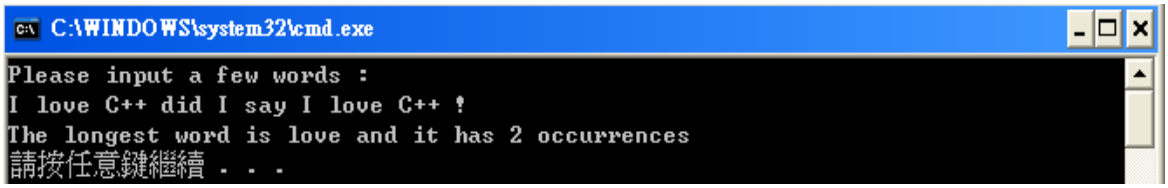
(1) In the first part, include the header file and write a program to allow users to enter some fractions from input (e.g., 2/3 3/4 2/5 and using # to terminate the input) then output the summation of these fractions (e.g., 109/60 in this case).

(2) Extend the program to again allow users to enter some fractions from input (e.g., 2/4 3/4 2/8 and using # to terminate the input) then output their corresponding reduced form (e.g., 1/2 3/4 1/4 in this case).



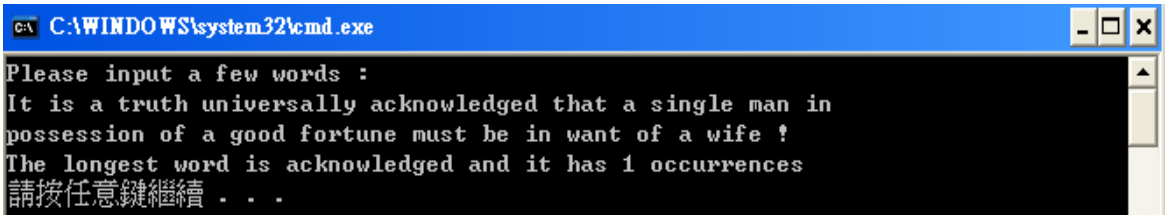
4. **(40%)** (a) Write a program to input a few words. Then output the longest word and its occurrence. For simplicity, assume that each word separates from each with a space and no punctuation existed. Use ! to signify the end of inputs. Below is a sample input sentence and sample run:

```
I love C++ did I say I love C++ !
```



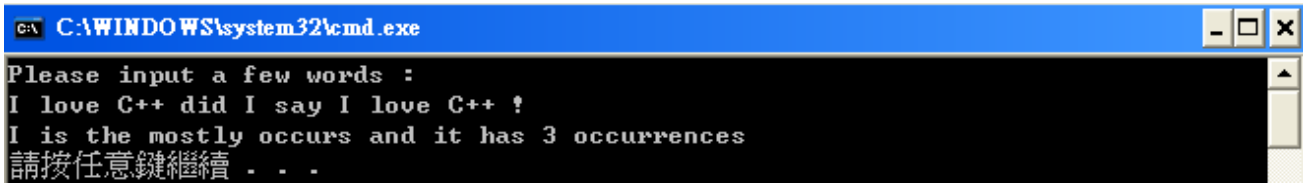
A more complex input looks like (from Pride and Prejudice (傲慢與偏見))

```
It is a truth universally acknowledged that a single man in
possession of a good fortune must be in want of a wife !
```

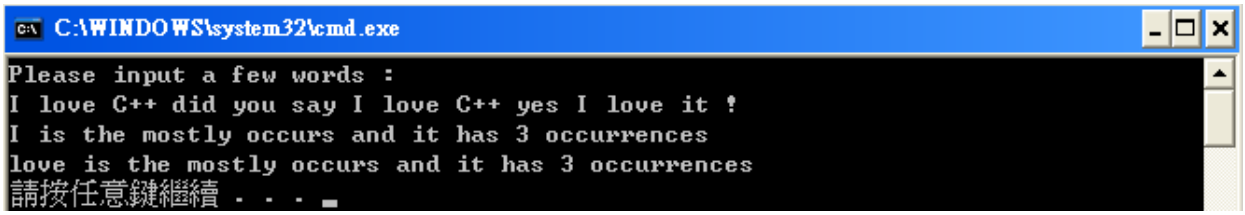


(b) Write a program to input a few words again. Then output the word(s) that mostly occurs and its occurrence (出現最多次的字與出現次數). For simplicity, assume that each word separates from each with a space and no punctuation existed. Use ! to signify the end of inputs. Below are two sample input sentences and two sample runs:

```
I love C++ did I say I love C++ !
```




```
I love C++ did you say I love C++ yes I love it !
```



HW Grading Policy:

1. You should consider about exception handling, e.g. error input, file opening fail, etc.
   請注意所有例外狀況的處理，例如：錯誤的符號字串輸入、檔案開啟失敗等。
2. The coding style includes your output format.
   輸出資料的格式將納入格式評分。
3. If your code is not compilable, your score in this problem is zero (including coding style).
   若程式無法編譯，則該題以零分計算。(包含格式分數)
4. Your program will be tested with other data which is not the same as provided samples.
   除了題目所提供的範例測試資料以外，作業程式碼將以額外的測試資料進行測試。
5. If tricky situations occur, the grade depends on Prof. Chen or TA's judgment.
   假如有特殊情況發生，則依據陳俊杉教授以及助教們的判斷給分。

- Coding Style (20%): 編碼格式分數
    1. format
       整體形式與輸出資料的格式
    2. comments
       註解
    3. readability
       可讀性
    4. variables naming
       變數命名方式
    5. typesetting
       型別設定

- Functionality (80%): 功能性分數

1. run-time performance:
   執行時的表現
   1) samples not passed -> x
      範例測資錯誤 => 此部分零分
   2) samples passed but some tests failed -> partial
      範例測資通過但是部分測資失敗 => 部份給分
   3) samples and tests all passed
      範例測資與所有測資通過 => 此部分滿分
3. excellent method++
   綜合以上，又以能展現解決問題的巧思尤佳。