

CS105C: Computer Programming C++ Programming Assignment #1

Adrian Trejo Nuñez
atrejo@cs.utexas.edu

Due: 11:59pm September 19, 2018

The purpose of this assignment is to familiarize yourself with writing basic C++ code. This assignment will cover simple I/O and writing functions.

1 Description

You will decrypt an easy, but not commonly used, cipher which encodes a secret message in seemingly normal text. Given the encrypted message, the most frequent character of each line is a character of the original message. Let's look at the following nonsensical text:

```
Unlike you I take umbrage with gnu!  
To think and code well tantalizes the mind.  
birds fly together and chirp.  
George Orwell was wrong about 1984.  
Can your cat code?  
Kick the kilos.  
snakes sneakily slither.
```

For this text, the output should be:

```
u 4  
t 5  
r 3  
o 4  
c 3  
k 3  
s 4
```

Your program should read text from standard input. For each line, you should print out the most frequent character and its frequency to standard output. All characters should be printed out in lower case. To keep things simple, we will ignore grammatical symbols and whitespace. If a line is blank or does not have any letters on it, then ignore it and move on to the next one. If a line has a tie for most appearances, use the lowest valued character in terms of ASCII value.

2 Directions and Hints

2.1 Input

Before you start to code the main logic, first make sure you are able to get the input task working. Your first step should be to get your program to read the input. There are many ways to read text in C++, but the simplest approach for this program uses `std::getline`:

```
std::string line;
while (std::getline(std::cin, line)) {
    // process the line
}
```

`std::getline` reads a single line from `std::cin` (standard input) until there is no more input, at which point it will return false and the loop will terminate. The variable `line` holds the contents of the current line inside the loop body.

The `std::string` class contains many functions to access the characters so please read the documentation for ideas or ask in class, Piazza, or office hours.

2.2 Counting letter frequency

You should write a function that takes a `std::string` or `char[]` and returns the most frequent letter character, or some special character if the line is blank or only has non-letters. You may find the following functions useful: `std::isalpha` and `std::tolower`. You can write a separate function that takes a `std::string` or `char[]` and a character, and returns the number of times the character appears in the line.

3 Submission and Grading

- You will submit a single .cpp file named `assignment1.cpp` on Canvas.
- Make sure your final submission compiles without any external dependencies other than the STL. I will compile it as:

```
clang++ --std=c++11 -o assingment1 assignment1.cpp
```

If your code does not compile, you will receive zero points for this assignment. The code you submit must compile on the lab machines in GDC, since they are identical to the machine I will use for grading.

- I will automate the grading process by running your program against several test cases (including the example in this PDF). Make sure the output of your program conforms to the format described in the description section. You should write several test cases yourself and check that your program outputs the correct solution.
- A small portion of your grade is style, so please make sure you comment your file and you adhere to good coding practices. In particular, you should be consistent throughout your code.
- If you have any questions about the assignment, please ask on Piazza.