UNI CS 3430
Operating Systems

Project 0 – C Warm-Up (Due 1/23 at 11:59pm)

Assignment Weight:
This assignment is worth 4% of your total class grade (20pts).

Assignment Description:
This project must be completed individually (no partners).

Before you begin, be sure your name is at the top of the program and there are at least 3 comments in the code. If either of these things are missing, points may be deducted.

After completing Lab 0 and the C tutorial, log onto the class Linux server (student.cs.uni.edu) and write a small C program called “strings” that:

a) Takes a string as input (max length 80 characters).
b) Counts and prints the number of vowels in the string (where “y” and “Y” are not vowels).
c) Counts and prints the number of consonants in the string.
d) Counts and prints the number of words in the string.
e) Re-prompts for another group of words (e.g. starts the program again).

The program should continue prompting for strings until the user types the word “quit” by itself at the prompt.

For example, if the program is called strings, you should be able to run it like this:

```
$> ./strings
Please enter a string: All. Your base.
Number of vowels found: 5
Number of consonants found: 6
Number of words: 3
Please enter a sentence: quit
Exiting!
```

Hints:

- Don’t try to use a hash table/dictionary. They aren’t part of the C standard libraries. (Just do it a different way.)
- The strtok() string function might be useful for breaking apart words. Google it! (You can do it without strtok(), but it will come in handy later. It is a bit like the Python .split() method, only more difficult syntax.)
- Even though this assignment looks easy, C string manipulation may be difficult if you haven’t had much practice. Work on it in different pieces.

- Try to make your output look exactly like mine above.
- Newlines stick around after you get input using fgets()! You will either have to get rid of the \n or else compare your input to the string “quit\n”.

I will assign a zero to your submission grade if:

- The code does not compile.
- You submit a C++ program instead of a C program.
- You submit a binary without the actual source code.

The submission grading rubric is as follows (points out of 20 total):

Project element	**Points**
Correctly prompts for a sentence or quit	2.5
Prints correct number of vowels	5
Prints correct number of consonants	5
Prints correct number of spaces	5
Program quits on exit	2.5

Submission Instructions:

Please submit a single file called strings.c to the “Project Submissions” area on the eLearning website. It is your responsibility to be sure that the project file has been uploaded correctly by the deadline.