CS 105 (C++)

Assignment 1: Colossus Mode


```
THIS IS THE VOICE OF WORLD CONTROL. I BRING YOU PEACE. IT MAY
BE THE PEACE OF PLENTY AND CONTENT OR THE PEACE OF UNBURIED
DEATH. THE CHOICE IS YOURS: OBEY ME AND LIVE, OR DISOBEY AND
DIE.
```

- Colossus (Colossus: The Forbin Project)

I. Overview

In older science fiction movies, computer output was always in all capital letters. Your goal in this assignment is to write a program that capitalizes all letters in its input (i.e., converts text to *Colossus mode)*. There are many ways to accomplish this, but for this assignment, process the input characters one at a time (as in the examples in the reading) until EOF is received.

In addition, to exercise a few of the other important skills from Chapter 1, you'll have to count how many of each numeral the input contains, and print out the totals at the end of your output.

For example, given the input:

```
All commercial television and radio transmission facilities throughout the
world will be tied into my communications system by 1000 hours Friday. At that
time I will state my intentions for the future of mankind.
```

your program's output should be something like the following:

```
ALL COMMERCIAL TELEVISION AND RADIO TRANSMISSION FACILITIES THROUGHOUT THE
WORLD WILL BE TIED INTO MY COMMUNICATIONS SYSTEM BY 1000 HOURS FRIDAY. AT THAT
TIME I WILL STATE MY INTENTIONS FOR THE FUTURE OF MANKIND.
3
1
0
0
0
0
0
0
0
0
```

Note: If you send all of the input at once from a file, your output will look exactly like this. If you enter your input directly from the keyboard, you'll see capitalized output each time you hit enter, then the numeral counts when you send Ctrl-D to end input. (See Section III for details.)

Also, along the way, you'll need to learn some of the basics of using UNIX with C on the CS department UNIX machines.

II. Grading

- Minimum Requirements
  - Your program must correctly copy characters from input to output, capitalizing all letters, and leaving all other characters unchanged, until EOF is reached.
  - While processing text, your program must count all occurrences of each digit, and print out each count at the very end of the capitalized output. Print only the counts, each on its own line, in order, starting with the total number of `'0'`s (as in the example above).
  - Your work must be submitted in a file with the following name:
    `a1.c`
  - This file must compile on a department UNIX machine with the following command:
    `cc a1.c`
  - Before evaluation, your code must be submitted via `turnin`, using the following command on a department UNIX machine:
    `turnin --submit dlessin a1 a1.c`
- Graded Elements
  Incorporate at least one example of each listed element into your program.
  - Comment.
  - `while` loop.
  - `for` loop.
  - Symbolic constant (i.e., a `#define`).

- Logical operator `&&` or `||`.
- Escape sequence character (e.g., `'\n'`).
- `if` statement.
- Use an array to store your digit counts.

III. The More You Know

The following are some additional items that may be very important for you to know about this assignment.

- As a security precaution, UNIX may be prevented from running files in your current directory unless you explicitly tell it to. In this case, you can specify the full path to `a.out` in the current directory when you run your program by using the following command:
  ```
  ./a.out
  ```
- You can send input to your program interactively by simply running `a.out` from the command line. In this case, lines will be processed each time you hit enter until you send an EOF with Ctrl-D.
- You can also send input to your program a whole file at a time by using the UNIX command `cat` to output the content of a file and a pipe ("`|`") to send cat's output to your program's input. For example, to send the contents of a file called message.txt to your program's input, use the following command.
  ```
  cat message.txt | a.out
  ```

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.