1 Objectives

The objectives of this project are to get you more familiar with:

1. The programming environment
2. Working with the basics of the C language (e.g., syntax, types, strings, loops, displaying, etc.)
3. Working with command-line arguments via `argc` and `argv`
4. Reading and understanding `man` pages
5. Converting strings to numbers using the `strtol` library function

2 Requirements

Big picture: write a program called `proj1` that takes a string and a number as inputs (i.e., passed via argv), and then redisplays the string with reverse capitalization, and then displays all the numbers from 1 to the input number (inclusive).

2.1 Interface Requirements

1. The program shall be written in a file called `proj1.c`.
2. `proj1` shall take two strings as input (e.g., "`./proj1 Hello 8`"). The first input string can be any length, while the second input string is expected to represent an integer greater than 0 but less than 25.
3. `proj1` shall detect when the input is invalid.
   When `proj1` detects invalid input, it shall display a meaningful error message, and then exit with a value of 1. There are a few ways that input can be invalid, such as:
   a. Too many or too few arguments were provided on the command-line.
   b. The second input represents a number less than 1 or greater than 24.
   c. The second input does not represent a valid number (e.g., abc and 1ab).
4. After verifying that the input is valid, the program shall then do the following:
   a. Display the first input string with the capitalization reversed. For example, if "Hello" is the first input string, then the program shall display "hELLO".
   b. Display each number from 1 to the value of the second input string, separated by commas and spaces. For example, if "8" is the second input string, then the program shall display "1, 2, 3, 4, 5, 6, 7, 8".
5. An example of the required input and output is given below (where the '$' is the command-line prompt):

```
$ ./proj1 Hello 8
hELLO
1, 2, 3, 4, 5, 6, 7, 8
$
```

6. If no input errors are detected, then the program shall display the required output and then exit with a value of 0.

3 Submission

Before the deadline, post to Sakai a file called `proj1.tar`, which you can create by doing the following on the command-line:

```
tar –cvf proj1.tar Makefile proj1.c
```

Late submissions will not be accepted. If you are not done by the due date, then turn in what you have for partial credit (instead of getting a zero for turning in nothing).

4 Grading

Grading will be based on the following **guidelines**:

1. The code compiles without errors when using the gcc compiler (20).
2. The code compiles without warnings when the -Wall compiler option is used with the gcc compiler (10).
3. No segmentation faults or other crashes occur when the code is run (10).
4. The code handles valid input as specified in Section 2. In particular, the following tests of valid input will be performed (20):
   a. `./proj1 Hello 8`
   b. `./proj1 WORLD 10`
   c. `./proj1 WoW! 20`
   d. `./proj1 thisisaTest 24`
5. The code handles **in**valid input as specified in Section 2. In particular, the following tests of invalid input will be performed (20):
   a. `./proj1`
   b. `./proj1 hello`
   c. `./proj1 hello world`
   d. `./proj1 hello world goodbye`
   e. `./proj1 Jan 3rd`
   f. `./proj1 feb 3g`
   g. `./proj1 Hello 0`
   h. `./proj1 Hell0 -1`
   i. `./proj1 Hello 25`
   j. `./proj1 World 30`
6. The `strtol` library function was used to convert the input string to a number. (10)
7. Code complies with the style guide. (10)
8. Deductions may occur in the following situations:
   a. The code appears to be hard-coded to give the correct answers to the test input.

b. You have 24 `if` statements to determine what the output should be (and other such obvious inefficient programming).
c. I reserve the right to include other reasons I have not yet encountered.

5 Advice and Reminders

1. **See the "Tips" link at the top of the main Wiki page**.
2. You may find the following library functions useful: `isupper`, `islower`, `isdigit`, `toupper`. See their man pages.
3. Remember that the `strlen` function can be used to find the length of a string, and the `strcmp` function is used to compare two strings for equality. See their man pages.