CS 100 Project Three – Fall 2017

Project Overview: This project takes a stream of text (strings) and formats the text into a set of lines with a specific width. However, rather than just do the standard left-justified text that you see in this paragraph, we take the approach used by Leonardo Di Vinci and generate a "mirror" output of the text. That is, we print the text from right-to-left rather than left-to-right. The program prompts the user for the length of an output line. The program then uses this as the width of all output lines. We right justify each line, as this is a true mirror of a standard left-justified paragraph.

We cannot do a true mirror of the text, as we do not have the ability to change the direction of our letters. We cannot generate **ɘ** instead of **e** (and so on). So we will just print the text out backwards. As an example, consider the input: **The quick brown fox jumps over the lazy old dog.**

This text, written backwards, is **.god dlo yzal eht revo spmuj xof nworb kciuq ehT**

Two sample executions of the program are shown below. The bottom row (black numbers) is added to show you there are that many columns, you do **not** have to generate that line of numbers as part of your output. Note that all lines are right-justified.

```
./a.out
Enter the output line length: 10
Enter your text (control-d to exit)
The quick brown fox jumps over the
lazy old dog.
<control-d>

 kciuq ehT
 xof nworb
revo spmuj
  yzal eht
  .god dlo
1234567890
```

```
./a.out
Enter the output line length: 25
Enter your text (control-d to exit)
The quick brown fox jumps over the
lazy old dog.
<control-d>

spmuj xof nworb kciuq ehT
   .god dlo yzal eht revo
1234567890123456789012345
```

Two quick comments on this program:

- First, you will never have a case where a single word (string of text) in the input is longer than the size of the output line specified by the user.
- Second, the longest output line that a user can specify is 100 characters in length. The smallest output line that can be specified is 10 characters in length.

What You Need To Do

- Create a directory **project3** on your machine. In that directory, create a file named **mirror.c**
- In **mirror.c**, write the code needed to implement this program. Make sure that you:
  - Include a header block of comments with your name and a brief overview of the program.
  - Prompt the user for the **width** of your output line.
  - Read the user's text (a set of strings) from standard input until the end-of-file, printing the "mirrored" words in lines of **width** characters
- When you are ready to submit your project, bundle your **project3** directory into a single (compressed) zip file. See the **Basics** document on Blackboard if you do not remember how to do this.
- Once you have a compressed zip file that contains your **project3** code, submit that file to Blackboard.

Project 3 is due at 5:00pm on Friday, October 6. Late projects are not accepted.