Program Submission Instructions:

- You must submit your source code file
- The source code file must be submitted in Webcourses from the assignment page
- All source code must be in exactly one file of type .c, .cpp, or .java

CIS 3360 – Security in Computing
Fall 2014
Program #1 Rail Fence Cipher (100 points)

In this assignment you'll write a program that encrypts the alphabetic letters in a file using the Rail Fence cipher where the fence depth can be any size from 2 up to 25. Your program will take two command line parameters containing the names of the file storing the encryption key and the file to be encrypted. The program must generate output to the console (terminal) screen as specified below.

Command Line Parameters

1. Your program compiles and run from the command line.
2. The program executable must be named "railcipher" (all lower case, no spaces or file extension).
3. Input the required file names as command line parameters. Your program may NOT prompt the user to enter the file names. The **first parameter** must be the name of the encryption key file, as described below. The second **parameter** must be the name of the file to be encrypted, as also described below. The sample run command near the end of this document contains an example of how the parameters will be entered.
4. Your program should open the two files, echo the input to the screen, make the necessary calculations, and then output the ciphertext to the console (terminal) screen in the format described below.

Encryption Key File Format

The encryption key file will store a single positive integer, *n (1 < n < 25)*, on the first line, indicating the depth of the rail fence.

Format of the File to be Encrypted

The file to be encrypted can be any valid text file with no more than 9991 letters in it. (Thus, it's safe to store all characters in the file in a character array of size 10000, including any padding characters.) Please note that the input text file will also generally have punctuation, numbers, special characters, and whitespace in it, which should be ignored. You should also ignore whether a letter is uppercase or lowercase in the input file. Thus, you should treat 'A' and 'a' the same in your program.

Output Format

The program must output the following to the console (terminal) screen:

1. Echo the input key file
2. Echo the input plaintext file
3. Ciphertext output produced from running the cipher against the input plaintext file.

The ciphertext output portion should should consist of only lowercase letters in rows of exactly 80 letters per row, except for the the last row, which may possibly have fewer. These characters should correspond to the ciphertext produced by encrypting all the letters in the input file. Please note that only the alphabetic letters in the input plaintext file will be encrypted. All other characters should be ignored.

What to Turn In to WebCourses

You must submit this assignment in the form specified at the top of this assignment.

Program Notes and Hints

Your program must read in an input plaintext file that may contain uppercase letters, lowercase letters and non-letter characters. Your program must distinguish between these three groups so that only the letters get encrypted. All non-letter characters in the file are simply skipped and not counted as part of the plaintext. Please note that although both upper case and lower case letters will be encrypted, your program should not treat them differently, that is, the program should process an upper case input letter the same as the corresponding lower case letter, i.e., it should treat an 'A' the same as an 'a'.

One possible breakdown to solve this problem is as follows:

1) Write a section of code or function that reads only the upper and lower case letters in the input file into an char array of size 10000, storing only the appropriate lowercase letters in the character array.

2) Write a section of code or function that takes as input the array from section 1 and the encryption key and produces an array of ciphertext storing only lowercase letters.

3) Write a section of code or function that takes as input the array storing the ciphertext and outputs it to the screen in the format specified. Additional functions or code will be needed to echo the input key and plaintext files.

Sample Key File

```
5
```

Sample Input File

```
CS: the science that deals with the theory and methods of processing information in
digital computers, the design of computer hardware and software, and the
applications of computers.

IT: the development, implementation, and maintenance of
computer hardware and software systems to organize and communicate information
electronically. Abbreviation:  IT

Computers are man-made tools that aid us in solving other problems. A biologist is
trying to figure out how life works, physicists and chemists are trying to figure out
how items react in our universe, mathematicians are trying to figure out rules for
man-made systems.

Any research problem that may improve a computer's capability of helping solve a
problem or any research problem that sheds light about a new way to do something with
a computer is part of CS.

CS students tend to find jobs where they program at least some.
In the process, they are solving problems.
Challenges: It's impossible to teach all the new languages/toys.
Ultimately, we just need to teach our students how to think, so that they can pick up
new things on their own. Our biggest challenge is getting them to buy into that.

Ethical: Lots, with security etc.
```

Corresponding Output File

```
The output file is pending. It will be updated soon. (IT WILL BE A BLOCK OF LOWERCASE
TEXT – 80 Columns wide.
```

Sample Run Command

C or C++ program:

```
prompt> ./railcipher samplekey.txt sampleplain.txt
```

Java program:

```
prompt> java railcipher samplekey.txt sampleplain.txt
```

Grading Rubric

The total possible score for this program is 100 points. The following point values will be deducted for the reasons stated:

[-100 points] Your program does not successfully compile from the command line with one of these commands:

C program:	prompt>	gcc –o railcipher [your_file_name].c
C++ program:	prompt>	g++ –o railcipher [your_file_name].cpp
Java program:	prompt>	javac railcipher.java

Note: If you are submitting a Java program, the class file must be named "railcipher.java" and the class name must be "railcipher".

[-90 points] Your program does not run from the command line without error or produces no output.

[-70 points] The program compiles, runs, and outputs the key matrix and input file, but crashes thereafter or produces no encryption output.

[-50 points] The program compiles, runs, echoes the inputs, and generates encryption output, but the encryption output is incorrect (ignoring case) and it is not formatted correctly (not all letters or not all lowercase or not 80 letters per line).

[-25 points] The program compiles, runs, echoes the inputs, generates encryption output, and the encryption output is correct (ignoring case), but it is formatted incorrectly (not all letters or not all lowercase or not 80 letters per line).

[-25 points] The program compiles, runs, echoes the inputs, and generates encryption output, but the encryption output is incorrect (ignoring case) although it is formatted correctly (all lowercase letters, 80 letters per line).

[no deductions] The program compiles, runs, echoes the inputs, generates encryption output, the encryption output is correct (ignoring case), and it is formatted correctly (all lowercase letters, 80 letters per line).