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 2020
Program #1 Hill Cipher (100 points)

In this assignment you'll write a program that encrypts the alphabetic letters in a file using the Hill cipher where the Hill matrix can be any size from 2 x 2 up to 9 x 9. 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 **must** compile and run from the command line on the CECS student server running *Ubuntu* named ***Eustis*** (available at ***eustis.eecs.ucf.edu***).
2. The program executable must be named “hw1” (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.

Note: If the plaintext file to be encrypted doesn't have the proper number of alphabetic characters, pad the last block as necessary with the lowercase letter '**x**'. (*Yes, this is different from what is in our class slides, but it is necessary for us to do this so we can know what outputs to expect for our test inputs.*)

Encryption Key File Format

The encryption key file will store a single positive integer, *n* $(1 < n < 10)$, on the first line, indicating the number of rows and columns in the encryption matrix. The following *n* lines will contain the contents of each row, in order, of the encryption matrix, separated by spaces.

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. This is best accomplished by converting all uppercase letters to lowercase. Remember to discard the space character and all other **non-alphabetic** characters when building your plaintext input buffer.

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 consist of only lowercase letters in rows of exactly 80 letters per row, except for 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 over 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
```
3
1 1 6
3 3 1
5 2 7
```

Sample Input File
```
Turing played a crucial role in cracking intercepted coded messages
that enabled the Allies to defeat the Nazis in many crucial
engagements, including the Battle of the Atlantic, and in so doing
helped win the war.
```

Corresponding Output File
```
Key matrix:

1  1  6
3  3  1
5  2  7


Plaintext:

turingplayedacrucialroleincrackinginterceptedcodedmessagesthatenabledthealliesto
defeatthenazisinmanycrucialengagementsincludingthebattleoftheatlanticandinsodoin
ghelpedwinthewarx


Ciphertext:

leufreaatujtaxtswejylxbqhncdbvspbodhvioshtpuuizhbhgagaewobovrjotouhwgyehzsvqcszy
ohfgdypluevfcanpmxvyrjswejlytfzeqsbsxanpdhqluspenfpwavcwhfyehhqlxgdkesmexldrnhvj
mwbirjryferndpzwn
```

Sample Run Command

```
C or C++ program:

prompt> ./hw1 k1.txt p1.txt

Java program:

prompt> java hw1 k1.txt p1.txt
```

Source code comment blocks (REQUIRED)

The **header** of the source code file should contain the following comment block:

```
/*============================================================================
|  Assignment:  HW 01 – Encrypting a plaintext file using the Hill cipher in the key file
|
|   Author:  Your name here
|   Language:  Java
|
|  To Compile:  javac Hw01.java
|
|  To Execute:  java Hw01 hillcipherkey.txt plaintextfile.txt
|                 where the files in the command line are in the current directory.
|                The key text contains a single digit on the first line defining the size of the key
|                while the remaining lines define the key, for example:
|                 3
|                 1 2 3
|                 4 5 6
|                 7 8 9
|                 The plain text file contains the plain text in mixed case with spaces & punctuation.
|
|  Class:  CIS3360 - Security in Computing - Fall 2020
|  Instructor:  McAlpin
|  Due Date:  per assignment
|
 +=============================================================================*/
```

The last lines of your code should contain the following **Academic Integrity** statement:

```
/*============================================================================
|    I [your name] ([your NID]) affirm that this program is
| entirely my own work and that I have neither developed my code together with
| any another person, nor copied any code from any other person, nor permitted
| my code to be copied  or otherwise used by any other person, nor have I
| copied, modified, or otherwise used programs created by others. I acknowledge
| that any violation of the above terms will be treated as academic dishonesty.
 +=============================================================================*/
```

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 hw1 [your_file_name].c
C++ program:    prompt>     g++ –o hw1 [your_file_name].cpp
Java program:   prompt>     javac hw1.java
```

Note: If you are submitting a Java program, the class file must be named "hw1.java" and the class name must be "hw1".

[-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).

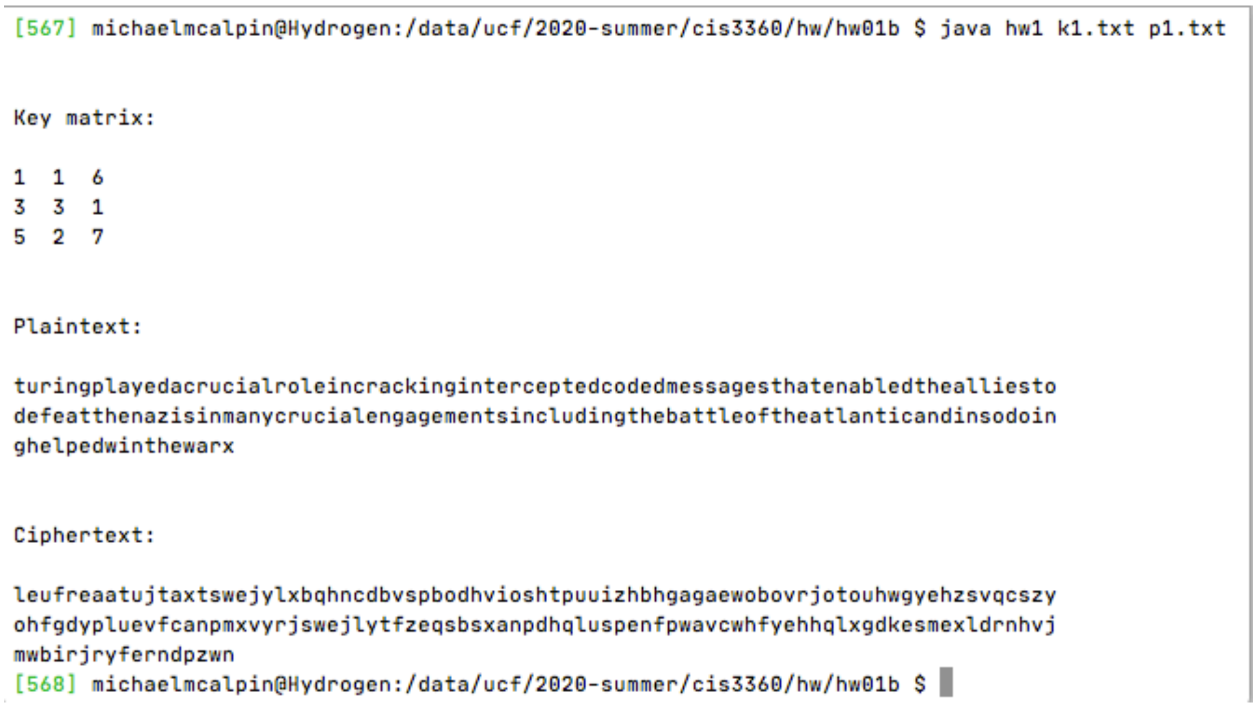
[-25 points] The program compiles, runs, echoes the inputs, and generates encryption output, but the encryption output is incorrect (ignoring case) although it is formatted correctly (all lowercase letters, 80 letters per line).

[-25 points] Missing **Academic Integrity** statement in source code file.

[-25 points] Missing **header** statement containing usage instructions in source code file.

[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).

Screen Capture of Programming Assignment 1
Sample Run



```
[567] michaelmcalpin@Hydrogen:/data/ucf/2020-summer/cis3360/hw/hw01b $ java hw1 k1.txt p1.txt


Key matrix:

1  1  6
3  3  1
5  2  7


Plaintext:

turingplayedacrucialroleincrackinginterceptedcodedmessagesthatenabledthealliesto
defeatthenazisinmanycrucialengagementsincludingthebattleoftheatlanticandinsodoin
ghelpedwinthewarx


Ciphertext:

leufreaatujtaxtswejylxbqhncdbvspbodhvioshtpuuizhbhgagaewobovrjotouhwgyehzsvqcszy
ohfgdypluevfcanpmxvyrjswejlytfzeqsbsxanpdhqluspenfpwavcwhfyehhqlxgdkesmexldrnhvj
mwbirjryferndpzwn
[568] michaelmcalpin@Hydrogen:/data/ucf/2020-summer/cis3360/hw/hw01b $ 
```