Programming Assignment 1
Hill Cipher

Michael McAlpin

June 2, 2023

1 Hill Cipher

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 $2x2$ up to $9x9$. 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.

1.1 Command line parameters

1. Your program **must compile** and **run** from the **terminal command line**.
2. 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, also described below. The sample run command near the end of this document contains an example of how the parameters should be entered and processed.
3. Your program should open the two files, echo the processed input to the screen, make the necessary calculations, and then output the ciphertext to the console (terminal) screen in the format described below.
4. Your program **must** be named pa01 for whichever language you choose. For example, **pa01**.c, **pa01**.cpp, **pa01**.java, **pa01**.py, or **pa01**.go.

Note
If the plaintext file to be encrypted doesn't have the proper number of alphabetic characters to *match* the key size, pad the last block as necessary with the lowercase letter **x**. Make sure that all the input characters are lower case only.

1.2 Formats

1.2.1 Encryption Key File Formats

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

1.2.2 Encryption Plaintext File Formats

The file to be encrypted can be any valid text file with no more than $9,991$ letters in it. (Thus, it's safe to store all characters in the file in a character array of size $10,000$, 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 convert uppercase letters to lowercase in the input file, correspondingly lowercase letters do not need to be converted. Thus, the program will treat **A** and **a** the same in your program. Remember that the input plaintext file may need to be *padded to match the block size* of the key.

1.2.3 Output Format

The program must output the following to the console (terminal) screen, also known as `stdout`:

1. Echo the numbers from the input key file.
2. Echo the lowercase alphabetic text derived from the input plaintext file.
    - Remember to pad with **x** if the processed plaintext does not match the block size of the key.
3. Ciphertext output produced from encrypting the input key file against the input array specified in the key file.

The output portion of the input plaintext file 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 using the numbers collected from the input key file and applied as a Hill cipher, via matrix multiplication (see additional notes and pseudocode at the end of the document). Please note that only the alphabetic letters in the input plaintext file will be encrypted. All other characters should be ignored.

1.2.4 Program Execution

The program, **pa01**, expects two inputs at the command line.

- The first parameter is the name of the key file,which contains a single positive integer, n where $(1 < n < 10)$, 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.
- The second parameter is the name of the plaintext file, note that this input files may contain non-alphabetic characters (numbers, punctuation, white spaces, etc.). The valid inputs, as discussed previously, may be any alphabetic character, and should be converted to lower case.

1.2.5 Program execution - example

Note that the commands below are also outlined later on in this document for either **c**, **c++**, **Java**, **Go**, or **Python**. Also note that the first parameter is the *key* filename and the second parameter is the *plaintext* filename.

```
systemPrompt$ gcc -o pa01 pa01.c
systemPrompt$ ./pa01 kX.txt pX.txt
```

A sample program execution with outputs is shown below. It is explained in more detail later on in the Section *Sample inputs and outputs* on page 6.

```
systemPrompt$./pa01 k1.txt p1.txt

Key matrix:
   2   4
   3   5

Plaintext:
notonlyistheuniversestrangerthanwethinkitisstrangerthanwecanthinkwernerheisenber
gx

Ciphertext:
efqxsqciitepovwzytawitizyrytooaniiooqlassteocmancmgqovktqwanooqlekytqhkioaawesyt
ad
```