

Mixed Alphabet Cipher

Description

This software application encrypts and decrypts messages using the Mixed Alphabet Cipher's method. For more information on this method:

<http://crypto.interactive-maths.com/mixed-alphabet-cipher.html>

Files enclosed for demonstration.

The folder containing this program in the repository has two text files. The file plainText.txt contains the unencrypted message, while encrypted.txt is the encrypted version of the plain-text message. The keyword for the encryption/decryption of these messages is *edgar*.

User's Guide

In order to execute the program it is necessary to compile and run it through command line. For executing it, it is necessary to input three arguments through command line. The first one is the *keyword* of the encryption/decryption and the second one represents the command to tell the program to encipher or decipher the message. The second argument can be "encipher" or "decipher". The third argument represents the name of the text file that wants to be encrypted or decrypted

Input example:

```
GQYOT NCQEO FYFES DRRME LEINQ RKRLR MTHMB HGTHN MNVRQ TFRKE
STTWN FUMAQ RAYRE QSART RGTHV RBHGT HNMFE SDRRM EOEQT HGUKE
QKYBR QTHKR CQNUM ABNQT FHSOK NTRKR LRMTS TEQTH MCWHT FRACE
QEKKE MONRS TFRCN KADUC
```

Output example:

Input Message:

```
GQYOT NCQEO FYFES DRRME LEINQ RKRLR MTHMB HGTHN MNVRQ TFRKE
STTWN FUMAQ RAYRE QSART RGTHV RBHGT HNMFE SDRRM EOEQT HGUKE
QKYBR QTHKR CQNUM ABNQT FHSOK NTRKR LRMTS TEQTH MCWHT FRACE
QEKKE MONRS TFRCN KADUC
```

Output Message:

```
CRYPT OGRAP HYHAS BEENA MAJOR ELEME NTINF ICTIO NOVER THELA
STTWO HUNDR EDYEA RSDet ECTIV EFICT IONHA SBEEN APART ICULA
RLYFE RTILE GROUN DFORT HISPL OTELE MENTS TARTI NGWIT HEDGA
```

Program's Design.

Important Data Structures and Global Variables.

The program uses two HashMaps that represent the alphabet mapping of the encryption. One HashMap represents the text-to-cipher table where the each text character is the key, and the other one represents the cipher-to-text table where each cipher character is the key. The program also stores the input and output of the program by using two String variables called input and output respectively.

Methods.

keyPlacement().

This method is responsible for setting up the mapping of the keyword in the text-to-cipher mapping.

generateTextToCipherTable().

This method uses keyPlacement() to do the mapping of the keyword and produces the mapping of the characters that have not been associated between the text and cipher alphabets.

generateCipherToTextTable().

This method runs generateTextToCipherTable() in order to produce the text-to-cipher mapping. Then it uses this table in order to generate the HashMap that would invert the axis of the relationship. In other words, it makes the cipher alphabets the keys of the HashMap.

encipherToken().

The method takes a string and enciphers it using the text-to-cipher mapping. It also takes the string and stores it in the input variable, while storing the result of the encryption in the output variable.

decipherToken().

The method takes a string and deciphers it using the cipher-to-text mapping. It also takes the string and stores it into the input variable, while storing the result of the decryption in the output variable.

main().

The main reads the text of the input file and decrypts/encrypts every line of it using encipherToken() or decipherToken(). At the end, it prints the input and outputs of the processes by using the mentioned global variables. Furthermore, it is also necessary to mention that this method deals with possible three invalid arguments from the command line, by showing a message about the error and ending the program.