Programming Exercise: Known Language and Key Length

Assignment: English Language, Known Key Length

The first step of this mini-project is for you to write a program that breaks a Vigenère cipher, where you know that the language is English and the key length is 4.

We have provided three Java classes to get you started:

- **CaesarCipher**: This class provides an implementation of the Caesar cipher algorithm that you learned about earlier with public **encrypt** and **decrypt** methods. A few adjustments have been made to the code to make it easier for you to work with here:
  - This code follows the second, object-oriented design you learned about, in which the constructor takes the key.
  - The code provides public methods to encrypt or decrypt one single character **encryptLetter** and **decryptLetter**.
  - In the constructor, the alphabets are built to have upper- and lowercase letters to preserve the case of a message.
  - You should test this code in a tester class that creates a CaesarCipher object and attempts to encrypt and decrypt an entire message (such as **titus-small.txt**), as well as individual characters.
- **CaesarCracker**: This class provides an implementation of the Caesar cipher cracking (or breaking) algorithm that you learned about earlier. As with the **CaesarCipher** class, a few adjustments have been made:
  - The constructor takes a parameter for the most common letter, so it can be used for languages other than English.

  - Finding the key has been separated from decrypting the message. You can use the method **getKey** to pass in an encrypted message and receive the key back.
  - You can test these methods in the tester class by making a new CaesarCracker object and decrypting the file **titus-small_key5.txt** using no arguments for the constructor (default most common character ‘e’), and the file **lusiadas_key17.txt**, noting that the most common character in Portuguese is ‘a’.
- **VigenereCipher**: This class implements a Vigenère cipher. It has the following functionality:
  - `public VigenereCipher(int[] key)`: the constructor, which takes a key, which is an array of integers and initializes the field **ciphers**, which is an array of CaesarCipher objects.
  - `public String encrypt(String input)`: a method that encrypts the String passed in and returns the encrypted message.
  - `public String decrypt(String input)`: a method that decrypts the String passed in and returns the decrypted message.
  - `public String toString()`: returns a String representing the key for this cipher.
  - You can test these methods in the tester class by creating a VigenereCipher object with the key “rome”, which is {17, 14, 12, 4} in integers and encrypting and decrypting the file **titus-small.txt**, the encrypted first line of which is “Tcmp-pxety mj nikhqv htee mrfhtii tyv”.

Your first step in this mini-project is to write the three methods in the **VigenereBreaker** class. Specifically you should do the following:

- Write the public method **sliceString**, which has three parameters—a String **message**, representing the encrypted message, an integer **whichSlice**, indicating the index the slice should start from, and an integer **totalSlices**, indicating the length of the key. This method returns a String consisting of every **totalSlices**-th character from **message**, starting at the **whichSlice**-th character.

  You can test your method on these examples:

  sliceString("abcdefghijklm", 0, 3) should return "adgjm"

  sliceString("abcdefghijklm", 1, 3) should return "behk"

  sliceString("abcdefghijklm", 2, 3) should return "cfil"

  sliceString("abcdefghijklm", 0, 4) should return "aeim"

  sliceString("abcdefghijklm", 1, 4) should return "bfj"

  sliceString("abcdefghijklm", 2, 4) should return "cgk"

  sliceString("abcdefghijklm", 3, 4) should return "dhl"

  sliceString("abcdefghijklm", 0, 5) should return "afk"

  sliceString("abcdefghijklm", 1, 5) should return "bgl"

  sliceString("abcdefghijklm", 2, 5) should return "chm"

  sliceString("abcdefghijklm", 3, 5) should return "di"

  sliceString("abcdefghijklm", 4, 5) should return "ej"

- Write the public method **tryKeyLength**, which takes three parameters—a String **encrypted** that represents the encrypted message, an integer **klength** that represents the key length, and a character **mostCommon** that indicates the most common character in the language of the message. This method should make use of the **CaesarCracker** class, as well as the **sliceString** method, to find the shift for each index in the key. You should fill in the key (which is an array of integers) and return it. Test this method on the text file **athens_keyflute.txt**, which is a scene from *A Midsummer Night's Dream* encrypted with the key "flute", and make sure you get the key {5, 11, 20, 19, 4}.

- Write the public method **breakVigenere** with no parameters. This void method should put everything together, so that you can create a new **VigenereBreaker** in BlueJ, call this method on it, and crack the cipher used on a message. This method should perform 6 tasks (in this order):
    a. Create a new FileResource using its default constructor (which displays a dialog for you to select a file to decrypt).
    b. Use the **.asString()** method to read the entire contents of the file into a String.
    c. Use the **tryKeyLength** method, which you just wrote, to find the key for the message you read in. For now, you should just pass 'e' for **mostCommon**.
    d. You should create a new **VigenereCipher**, passing in the key that **tryKeyLength** found for you.
    e. You should use the **VigenereCipher**'s **decrypt** method to decrypt the encrypted message.
    f. Finally, you should print out the decrypted message!

Test this method on the text file **athens_keyflute.txt**, using key length 5. The first line should be "SCENE II. Athens. QUINCE'S house."