Breaking the Vigenère Cipher

📗	Module Resources	10 min
▶	Introduction	2 min
▶	Known Language and Key Length	5 min
📗	Programming Exercise: Known Language and Key Length	10 min
⭐	**Practice Quiz:** Known Language and Key Length	2 questions
▶	Unknown Key Length	4 min
📗	Programming Exercise: Unknown Key Length	10 min
⭐	**Practice Quiz:** Unknown Key Length	4 questions
▶	Unknown Language	4 min
📗	Programming Exercise: Unknown Language, Unknown Key Length	10 min
⭐	**Quiz:** Breaking the Vigenère Cipher	7 questions
📄	Extend Your Program	10 min
📄	End of Module Survey	10 min

Assignment: Multiple Languages

Finally, it is time to expand your program so that it can crack messages in other languages. To accomplish this, you need to write two new methods and modify two methods you have already written.

Specifically, you should do the following:

- In the **VigenereBreaker** class, write the public method **mostCommonCharIn**, which has one parameter—a HashSet of Strings **dictionary**. This method should find out which character, of the letters in the English alphabet, appears most often in the words in **dictionary**. It should return this most commonly occurring character. Remember that you can iterate over a HashSet of Strings with a for-each style for loop.

- In the **VigenereBreaker** class, write the public method **breakForAllLangs**, which has two parameters—a String **encrypted**, and a HashMap, called **languages**, mapping a String representing the name of a language to a HashSet of Strings containing the words in that language. Try breaking the encryption for each language, and see which gives the best results! Remember that you can iterate over the **languages.keySet()** to get the name of each language, and then you can use **.get()** to look up the corresponding dictionary for that language. You will want to use the **breakForLanguage** and **countWords** methods that you already wrote to do most of the work (it is slightly inefficient to re-count the words here, but it is simpler, and the inefficiency is not significant). You will want to print out the decrypted message as well as the language that you identified for the message.

- Modify the method **breakForLanguage** to make use of your **mostCommonCharIn** method to find the most common character in the language, and pass that to **tryKeyLength** instead of 'e'.

- Modify your **breakVigenere** method to read many dictionaries instead of just one. In particular, you should make a HashMap mapping Strings to a HashSet of Strings that will map each language name to the set of words in its dictionary. Then, you will want to read (using your **readDictionary** method) each of the dictionaries that we have provided (Danish, Dutch, English, French, German, Italian, Portuguese, and Spanish) and store the words in the HashMap you made. Reading all the dictionaries may take a little while, so you might add some print statements to reassure you that your program is making progress. Once you have made that change, you will want to call **breakForAllLangs**, passing in the message (the code to read in the message is unchanged from before), and the HashMap you just created.

Test your program on the file **athens_keyflute.txt**. Notice that the correct key "flute" {5, 11, 20, 19, 4} is detected for English, French, Danish, German, and Dutch, due to 'e' being the most common character and the languages having cognates, but English has the greatest number of valid words for its decryption.

Link to FAQ page for this course: http://www.dukelearntoprogram.com/course3/faq.php

Programming Exercise - Unknown Langu…

✓ Complete

👍   👎   🚩