

Coursera

Catalog

Search catalog

Q

For Enterprise

Java

▼

◀ Back to Week 4

✕ Lessons

Prev

Next

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: English Language, Unknown Key Length

Now that you have broken Vigenère ciphers with a known key length, it is time to expand your program’s functionality to break Vigenère ciphers of unknown key length. You will do this by adding three methods to your **VigenereBreaker** class, and then modifying your **breakVigenere** method.

Specifically, you should do the following:

- In the **VigenereBreaker** class, write the public method **readDictionary**, which has one parameter—a FileResource **fr**. This method should first make a new HashSet of Strings, then read each line in **fr** (which should contain exactly one word per line), convert that line to lowercase, and put that line into the HashSet that you created. The method should then return the HashSet representing the words in a dictionary. All the dictionary files, including the English dictionary file, are included in the starter program you download. They are inside the folder called ‘dictionaries’.
- In the **VigenereBreaker** class, write the public method **countWords**, which has two parameters—a String **message**, and a HashSet of Strings **dictionary**. This method should split the message into words (use **.split(“\\W+”)**, which returns a String array), iterate over those words, and see how many of them are “real words”—that is, how many appear in the dictionary. Recall that the words in **dictionary** are lowercase. This method should return the integer count of how many valid words it found.
- In the **VigenereBreaker** class, write the public method **breakForLanguage**, which has two parameters—a String **encrypted**, and a HashSet of Strings **dictionary**. This method should try all key lengths from 1 to 100 (use your **tryKeyLength** method to try one particular key length) to obtain the best decryption for each key length in that range. For each key length, your method should decrypt the message (using VigenereCipher’s **decrypt** method as before), and count how many of the “words” in it are real words in English, based on the dictionary passed in (use the **countWords** method you just wrote). This method should figure out which decryption gives the largest count of real words, and return that String decryption. Note that there is nothing special about 100; we will just give you messages with key lengths in the range 1–100. If you did not have this information, you could iterate all the way to **encrypted.length()**. Your program would just take a bit longer to run.
- Finally, you need to modify your **breakVigenere** method to make use of your new code. The steps below describe what your **breakVigenere** method should do. They are similar to the steps from the previous portion of this project but have been updated: *new items are in italics*, [and old items that you should no longer do are in square brackets.]

- Create a new FileResource using its default constructor (which displays a dialog for you to select a file to decrypt).
- Use that FileResource’s **asString** method to read the entire contents of the file into a String.
- You should make a new FileResource to read from the English dictionary file that we have provided.* [Use the **tryKeyLength** method that you just wrote to find the key for the message you read in. For now, you should just pass ‘e’ for **mostCommon**.]
- You should use your **readDictionary** method to read the contents of that file into a HashSet of Strings.* [You should create a new VigenereCipher, passing in the key that **tryKeyLength** found for you.]
- You should use the **breakForLanguage** method that you just wrote to decrypt the encrypted message.* [You should use a VigenereCipher object to decrypt the encrypted message.]
- Finally, you should print out the decrypted message!
-

Test this method on the text file **athens_keyflute.txt**. The first line should be “SCENE II. Athens. QUINCE’S house”, and the key is “flute”, or {5, 11, 20, 19, 4}. This file contains 376 valid words out of 386 (total word count ignores whitespace).

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

Programming Exercise - Unknown Key Le...

✓ Complete

👍

👎

🚩