**Vigenère**

**tl;dr**

Implemente un programa que encripte mensajes usando cifrado Vigenère, al igual que abajo.

$ ./vigenere ABC

plaintext: HELLO

ciphertext: HFNLP

**Background**

El cipher Vigenère’s mejora sobre Caesar’s cipher al encriptar mensajes utilizando una secuencia de teclas (o, poniéndolo de otra forma, un keyword). En otras palabras, si *p* es un texto plano, y *k* es una tecla (ej., un string alfabético, por lo cual A representa 0, B representa 1, C representa 2, …​, y Z representa 25), luego cada letra, *ci*, en el ciphertext, *c*, es computado como:

*ci*=(*pi*+*kj*)mod26

Note que este cifrado usa of *kj* en oposición a solo *k*. Y si *k* es más corto que *p*, entonces las letras en *k* deben ser reutilizadas cíclicamente tantas veces como tome encriptar *p*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| En otras palabras, si Vigenère mismo desea decir HELLO a alguien confidencialmente, usando una keyword de digamos, ABC, el encriptará la H con una key of 0 (ej., A), la E con una key de 1 (i.e., B), y la primera L con una key de 2 (ej., C), a este punto quedándose sin letras en el keyword, y teniendo que re utilizar (parte de) el para encriptar la segunda L con una key de 0 (ej., A) de nuevo, y la O con una key de 1 (ej., B) de nuevo. Y así escribirá HELLO como HFNLP. | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |

**Especificación**

Diseñe e implemente un programa que encripte mensajes usando cifrado Vigenère.

* Implemente su programa en un archivo llamado vigenere.c en un directorio llamado vigenere.
* Su programa deberá aceptar un solo argumento de línea de comando: un keyword, *k*, compuesta enteramente por caracteres alfabéticos.
* Si su programa es ejecutado sin ningún argumento de línea de comando, con másde un argumento de línea de comando, o con un argumento de línea de comando que contenga cualquier carácter no alfabético, su programa deberá imprimir un mensaje de error (de su elección) y salir inmediatamente, con main retornando 1 (de este modo significando un error).
* De lo contrario, su programa deberá proceder a solicitar al usuario una string de texto plano, *p*, (con un aviso plaintext:) con lo cual entonces deberá encriptar de acuerdo al cifrado Vigenère con *k*, finalmente imprimiendo el resultado (antepuesto a ciphertext: y finalizando con una nueva línea) y saliendo con main retornando 0.
* Con respecto a los caracters en *k*, usted deberá tratar a A y a como 0, B y b como 1, …​ , y Z y z como 25.
* Su programa deberá aplicar cifrado Vigenère solamente a un caracter en *p* si ese caracter es una letra. Todos los otros caracteres (números, símbolos, espacios, sígnos de puntuación, etc.) deben ser impresos en la salida sin cambio. Además, si su código se va a aplicar a la *jth* caracter de *k* para el *ith* caracter de *p*, pero este ultimo resulta ser un character no alfabético, usted deberá esperar para aplicar ese *jth* caracter de *k* al próximo carácter en *p*; usted no deberá avanzar aún al siguiente caracter en *k*.
* Su programa deberá preserva el tipo de cada letra en *p*.

**[Walkthrough](http://docs.cs50.net/problems/vigenere/vigenere.html" \l "walkthrough)**

**[Usage](http://docs.cs50.net/problems/vigenere/vigenere.html" \l "usage)**

Su programa deberá comportarse como los ejemplos a continuación. Asumiendo que el texto subrayado es lo que algún usuario a introducido.

$ ./vigenere 13

Usage: ./vigenere k

$ ./vigenere

Usage: ./vigenere k

$ ./vigenere bacon and eggs

Usage: ./vigenere k

$ ./vigenere bacon

plaintext: Meet me at the park at eleven am

ciphertext: Negh zf av huf pcfx bt gzrwep oz

**Probando**

Para ayudarle a probar vigenere, hemos escrito para usted un programa llamado devigenere que también recibe uno y solo un argumento de línea de commando ( keyword) pero cuyo trabajo es tomar el ciphertext como entrada y producir texto plano como salida. Para utilizar nuestro programa, ejecute:

~cs50/pset2/devigenere k

En su prompt, dónde k es una keyword. Presumiblemente usted querrá pegar la salida de su programa como entrada a nuestro programa; deberá asegurarse por supuesto, de utilizar la misma key. Recuerde que usted no necesita implementar devigenere, solamente vigenere.

**[check50](http://docs.cs50.net/problems/vigenere/vigenere.html" \l "code-check50-code)**

check50 cs50/2017/x/vigenere

**[Staff’s Solution](http://docs.cs50.net/problems/vigenere/vigenere.html" \l "staff-s-solution)**

~cs50/pset2/vigenere

[**Consejos**](http://docs.cs50.net/problems/vigenere/vigenere.html#hints)

¿No está Seguro de dónde empezar? Por suerte, este programa es bastante similar a [caesar](http://docs.cs50.net/problems/caesar/caesar.html)! Sólo que esta vez, uste necesita decidir cuál caracter en *k* usar al iterar de caracter a caracter en *p*.

**FAQs**

*¡Ninguna hasta ahora! ¡Recargue esta página periódicamente para revisar si llega alguna!*