83-CRIPT-vigenere

March 4, 2018

```
In [1]: alfb = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

In [2]: texto = "Through the use of abstraction and logical reasoning, mathematics developed fi

In [3]: print map(ord,[x for x in alfb])

[65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88


In [4]: print map(chr,map(ord,[x for x in alfb]))

['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S'


In [5]: from string import *
        def limpiar(texto,alfb):
            L = map(ord,[x.capitalize() for x in list(texto)])
            L1 = [item for item in L if item in map(ord,[x for x in alfb])]
            C1 = join(map(chr,L1),sep = "")
            return C1

In [6]: limpiar(texto,alfb)

Out[6]: 'THROUGHTHEUSEOFABSTRACTIONANDLOGICALREASONINGMATHEMATICSDEVELOPEDFROMCOUNTINGCALCULATI
```

Cifra de Vigenere

Puedes leer sobre este método de cifrado en Vigenere. Elegida una palabra clave, por ejemplo "CIRUELA", el método para encriptar consiste en, primero, obtener para cada letra de la clave el número ASCII que le corresponde:

```
In [7]: L3 = list("CIRUELA");L3

Out[7]: ['C', 'I', 'R', 'U', 'E', 'L', 'A']

In [8]: L4 = map(ord,L3);L4

Out[8]: [67, 73, 82, 85, 69, 76, 65]
```

Llamemos K a la lista de enteros correspondientes a la clave y m a la longitud de la clave, en nuestro ejemplo 7. Para cada resto módulo m, digamos k, tenemos un entero $K[k]$.

En el segundo paso del método de Vigenere, cada letra del mensaje, que ocupa una posición digamos N en el mensaje, se encripta de manera diferente según el valor del resto de dividir N entre m. Si el valor de ese resto es k, usamos k como clave para encriptarla mediante la cifra de César.

Finalmente, tenemos una lista de enteros, entre 0 y 255, y la transformamos en una cadena de caracteres, que son el mensaje encriptado.

En esta parte del ejercicio vamos a usar el número de orden en un alfabeto de 26 letras en lugar del código ASCII.

```
In [9]: texto2 = limpiar(texto,alfb);texto2

Out[9]: 'THROUGHTHEUSEOFABSTRACTIONANDLOGICALREASONINGMATHEMATICSDEVELOPEDFROMCOUNTINGCALCULATI

In [10]: len(alfb)

Out[10]: 26

In [11]: L_alfb = list(alfb);print L_alfb

['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S'


In [12]: L_alfb.index('D')

Out[12]: 3
```

Estas dos funciones sustituyen a *ord* y *chr* para el alfabeto de 26 letras que estamos usando.

```
In [13]: def ord2(c):
             return L_alfb.index(c)

In [14]: def chr2(n):
             return L_alfb[n]
```

Define una función para encriptar usando este método y otra para calcular la clave para desencriptar dada una clave para encriptar. Aplica las funciones a texto2 y comprueba que se recupera el texto original.

```
In [ ]:
```

Ataque sobre el método de Vigenere

La idea es probar longitudes crecientes de las claves, es decir, suponer que la longitud de la clave k es 1, o 2, o 3 ,etc. Para cada una de estas posibles longitudes dividimos el mensaje formando submensajes que tienen las letras cuya posición tiene un resto i respecto a k. cada uno de estos mensajes, si la longitud que estamos probando es la correcta, ha sido encriptado mediante César usando una clave distinta, pero podemos hallar la clave usando en análisis estadístico que sirve en el caso de César. De esa forma podríamos llegar a descubrir la clave, si tenemos suficiente cantidad de texto encriptado.

EJERCICIO: implementar este ataque, usando el texto copiado más abajo y la clave "CIRUELA".

him when the days of his fighting were full. Tell me, O Muse, of that ingenious hero who travelled far and wide after he had sacked the famo

```
In [15]: texto_largo
```

```
In [16]: print texto_largo
```

```
Tell me, O Muse, of that ingenious hero who travelled far and wide after he had sacked the famo
```

```
In [ ]:
```