Códigos y Criptografía - Curso 2019/2020

Práctica 1: Cifrado Afín. Cifrado César (caso particular del cifrado Afín)

- Mientras no se diga lo contrario, nuestro abecedario será

`abcdefghijklmnñopqrstuvwxyz'

- A cada letra le vamos a asociar un número:

a	b	c	d	e	f	g	h	I	j	k	l	m	n	ñ	o	p	q	r	S	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9																	

1.- Función numeros=letranumero (texto)

Construir una función para pasar las letras a números y poder operar con ellos.

Entrada: texto escrito en castellano y entre ' ' (con mayúsculas, ' ¿? …).

Salida: Vector numérico asociado al texto (una vez convertidas todas las letras en minúsculas y eliminado del texto plano todos los símbolos que no está en nuestro abecedario).

Ejemplo

```
>> numero=letranumero('Hola, buenos dias,¿empezamos?')
numero =
 7  15  11   0   1  21   4  13  15  19   3   8   0  19   4  12
 16   4  26   0  12  15  19
```

2.- Función cifradoafin=afin (clave, d ,texto)

Construir una función para cifrar un mensaje por el método Afín.

Entradas:

clave: Es al clave multiplicativa, el programa debe comprobar que es un número entero y con mcd(clave , 27)=1.

d: el desplazamiento, también tiene que ser un número entero.

texto: texto claro que queremos encriptar, una vez introducido (entre ' '), el programa debe eliminar todo lo que no sea de nuestro abecedario, para ello debe llamar a la función **letranumero(texto).**

Salida: El criptograma.

Ejemplo

>>cifradoafin=afin(14,7,'este metodo tambien es facil')

cifradoafin =jdqjnjqbvbqhnuljajdwhilz

3.- Función descifraafin=desafin (clave, d, texto)

Función para descifrar un mensaje que ha sido cifrado por el método Afín y conocemos las claves de cifrado (clave y d).

Entradas:

clave: el número entero que se ha usado para multiplicar.

d: el número entero que se ha usado para trasladar.

texto: texto encriptado, del que queremos obtener el texto claro.

Salida: El mensaje claro.

Ejemplo

>> descifraafin=desafin(14,7, 'jdqjnjqbvbqhnuljajdwhilz')

descifraafin =estemetodotambienesfacil

Ahora vamos a construir funciones encaminadas a realizar un análisis de frecuencias de las letras de un criptograma obtenido por un método de cifrado por sustitución carácter a carácter y monoalfabética.

Nos servirán para poder realizar el criptoanálisis y encontrar el mensaje original conocido el criptograma sin conocer las claves de cifrado.

4.- Función [frecuencia, freordenada]=cripto_ana_orden (v)

Función para obtener las frecuencias de cada letra de un criptograma obtenido por cifrado afín.

Entrada: criptograma, al que hemos llamado v (al ser texto introducir entre comillas).

Salidas:

frecuencia: es una matriz de 27x2 donde en la primera columna aparecen las frecuencias y en la segunda los números de las letras correspondientes a esas frecuencias

freordenada: es otra matriz de 27x2 que se obtiene ordenando de mayor a menor según la primera columna de la matriz *frecuencia.*

Ejemplo

v=‘ltbtrbnhyklrhshtjñhkljbfvtvsiyltvxbolyvhjvykhyslfhklshztvcvfhlzjyoioythkhshzwvyx bllzavlzbtaleavvhyhclyzombtjovthtrvzwyvnyhshzxbllzavfñhjoltkv'

>> [frecuencias,freordenada]=cripto_ana_orden(v)

frecuencias =

0.028369	0
0.056738	1
0.014184	2
0	3
0.0070922	4
0.028369	5
0	6
0.12766	7
0.014184	8
0.042553	9
0.042553	10
0.11348	11
0.0070922	12
0.014184	13
0.014184	14
0.042553	15
0	16
0	17
0.021277	18
0.042553	19
0.085106	20
0	21
0.11348	22
0.014184	23
0.021277	24
0.085106	25
0.06383	26

freordenada =

0.12766	7
0.11348	11
0.11348	22
0.085106	20
0.085106	25
0.06383	26
0.056738	1
0.042553	9
0.042553	10
0.042553	15
0.042553	19
0.028369	0
0.028369	5
0.021277	18
0.021277	24
0.014184	2
0.014184	8
0.014184	13
0.014184	14
0.014184	23
0.0070922	4
0.0070922	12
0	3
0	6
0	16
0	17
0	21

5.- Función comparo=barras (v)

Función que compara las frecuencias del criptograma con las letras en castellano, mostrando la comparación de dos formas: mediante tabla y mediante diagrama de barras.

Entrada: criptograma, al que hemos llamado v

Salida:

a: matriz de 27 x4 donde las columnas 1 y 3 son las frecuencias ordenadas de mayor a menor en castellano y en nuestro criptograma:

'% castellano' ' nº letra' '% criptograma' 'nº letra'

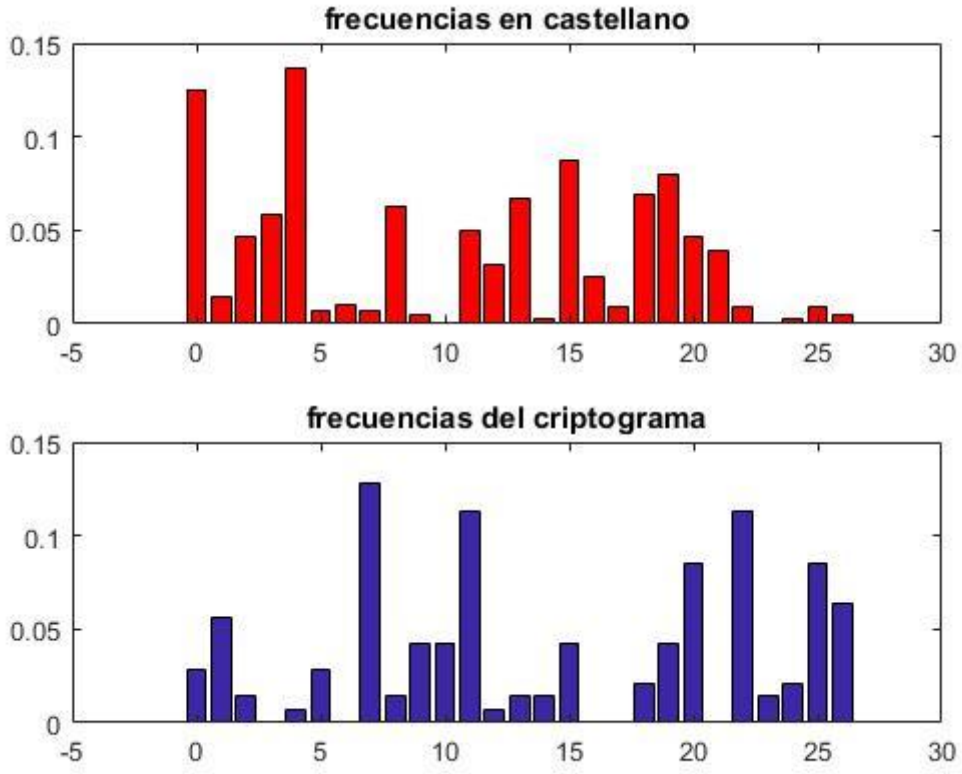
b: dos diagramas de barras. El primero indicando las frecuencias de las letras en castellano, el segundo indicando las frecuencias de cada letra en nuestro texto cifrado.

Ejemplo

v='ltbtrbnhyklrhshtjñhkljbfvtvsiyltvxbolyvhjvykhyslfhklshztvcvfhlzjyoioythkhshzwvyxbllzavlzbtaleavvhyhclyzombtjovthtrvzwyvnyhshzxbllzavfñhjoltkv'

```
>> comparo=barras(v)

table =

   'caste'    'Nª/letra'    'cifrado'    'Nª/letra'

comparo =

    0.1368         4         0.12766         7
    0.1253         0         0.11348        11
    0.0868        15         0.11348        22
    0.0798        19         0.085106       20
    0.0687        18         0.085106       25
    0.0671        13         0.06383        26
    0.0625         8         0.056738        1
    0.0586         3         0.042553        9
    0.0497        11         0.042553       10
    0.0468         2         0.042553       15
    0.0463        20         0.042553       19
    0.0393        21         0.028369        0
    0.0315        12         0.028369        5
    0.0251        16         0.021277       18
```

0.0142	1	0.021277	24
0.0101	6	0.014184	2
0.009	22	0.014184	8
0.009	25	0.014184	13
0.0088	17	0.014184	14
0.007	7	0.014184	23
0.0069	5	0.0070922	4
0.0052	26	0.0070922	12
0.0044	9	0	3
0.0031	14	0	6
0.0022	24	0	16
0.0002	10	0	17
0.0001	23	0	21



6.- Función inver=inv_modulo (A, m)

Función para calcular la inversa de una matriz de coeficientes enteros, modulo n.

Entradas:

A: una matriz; habrá que comprobar que los coeficientes son enteros y que es cuadrada porque vamos a calcular su inversa trabajando módulo *m*.

m: el módulo de trabajo.

Salida: la inversa de la matriz, módulo *m*, y en caso de no existir dicha inversa, un mensaje y el valor *inver =0*

.

Ejemplo

```
>> inver=inv_modulo([3 4.4;5 6],27)
esa matriz no tiene todos los elementos enteros
inver =  0
>> inver=inv_modulo([3 0;5 6],27)
la matriz tiene todos sus elementos enteros
la matriz no es inversible modulo27
inver = 0
>> inver=inv_modulo([3 2;5 6],27)
la matriz tiene todos sus elementos enteros
inver =
   21   20
   23   24
```

7.- Función criptoanalisis_afin (v, m)

Esta función realiza el criptoanálisis de un mensaje, que ha sido cifrado con afín, pero del que no conocemos las claves.

Entradas:

v: criptograma.

m: módulo de trabajo.

Salida:

Compara las máximas frecuencias, descifra el mensaje y nos dice que si queremos probar con otras claves. Si decimos que sí, pasa a las siguientes frecuencias y nos muestra el nuevo posible mensaje claro…….hasta que le digamos que no lo intente con otras claves porque ya tengamos el texto claro.

Al final debo tener el mensaje claro y el valor de las claves que se han usado para cifrarlo.

Ejemplo

v=’eymcklcdmgdcyescmeligvcqwbseiwycklevqgqwdgrlcldwveuiemwqwcrgvbmcipgevseiwycklevqgqwcscdelucgvnelyciwqzucl’

>>criptoanalisis(v,27)

a =

4 1

0 1

inva =

7 20

1 1

clave = 13

d =4

descifraafin =

amleoneclwcemaxelanswrebpgxaspmeonarbwbpcwznencpratsalpbpezwrglesdwarxaspmeonarbwbpexecantewrjanmespbkten

si quieres probar otra clave introduce 1, en caso contrario introduce 0 ……

…….

clave =14

d =2

descifraafin =

estapracticasehaterminadoyhemosaprendidocifrarconelmetodoafinytambienhemosaprendidoahacerlainversamodular

si quieres probar otra clave introduce 1, en caso contrario introduce 0 0