2022/23

Memoria práctica 1. Criptología. Cifrado Simétrico.

GRUPO 5.1.4
VÍCTOR CRIBEIRO PÉREZ
MANUEL FERNÁNDEZ FERREIRO
VÍCTOR GONZÁLEZ DEL CAMPO
DIEGO MOURE GONZÁLEZ

Índice

1) Instalen *Cryptool* 1 y realicen pruebas de cifrado y descifrado con los algoritmos vistos en clase de teoría.

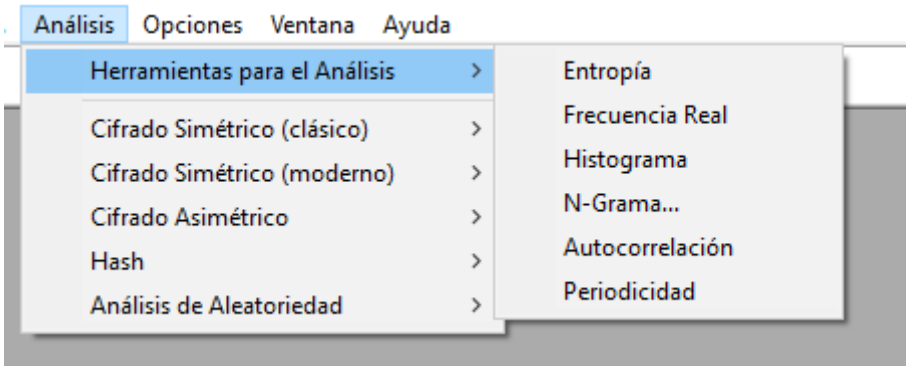
a. Busquen y documenten las opciones de *Cryptool* para el criptoanálisis de algoritmos de sustitución monoalfabeto

TEXTO CIFRADO (SUSTITUCIÓN MONOALFABETO):

PTHNPULUZLHOVYHJBHSZLYPHZBHZVTIYVJBHUKVHSKLZWLYAHYZLHKCPYAPHXBLFHUVLYHBUTBULJ
VKLTHKLYHZPUVXBLZLOHIHHJVUCLYAPKVLUBUUPUVJVTVAVKVZSVZKLTHZLJOHBUHVQLHKHHZBHSY
LKLKVYFLUCLGKLSHZOHIPABHSLZWHYLKLZKLWHQHKLSHJHIHUHCPVBUHIVUPAHOHIPAHJPHVUHTBL
ISHKHFHKVYUHKHJVUBUHZLUJPSSLGJHZPLSLNHUALFHSZHSAHYKLSHJHTHZLLUJVUAYHWYLWHYHKV
BUCLZABHYPVUBLCVBUNVYYVUBLCVFBUWHYKLIVAHZKLWPLSXBLLYHUBUCLYKHKLYVZBLHUVAHUW
YVUAVJVTVZLCPZAPHZLSLVJBYYPHTLALYSHZTHUVZLUSVZIVSZPSSVZFZHJHYBUWLXBLHUVWVYAHTV
ULKHZKLTHYMPSLULSXBLLZAHIHULZJYPAHZLZAHZWHSHIYHZHLSOHKHKLSVZJHILSSVZHGBSLZKLCBLS
CLHZBXBLYPKVWPUVJOVSVZJBHYLUAHJLUAHCVZFSLHNYHKLJLZBIBLUJVYHGHUHHIYPHLSWVYAHTV
ULKHZFLUCLGKLJBHYLUAHJLUAHCVZKLJVIYLLUJVUAYHJBHYLUAHTVULKHZKLVYVYLJPLUHJBUHKHZKL
ZWBHJZMBLHTPYHYZLHSLZWLQVFSLWHYLJPHXBLLYHVAYVFHUVCPVYLMSLQHKHSHOHIPABHSPTHN
LUKLBUHTHYPVULAHKLTHKLYHZPUVXBLCPVSHJHYHCPCHLPUALSPNLUALKLBUNBHWVJOPJVKLJHILSS
VZJHZAHUVZVQVZJLSLZALZFBUHZWLJAVHSLNYLFMLZAPCVJVTVBUHZWHZJBHZLUTLKPVKLAVKHZLZA
HZTHYHCPSSHZXBLZLZBJLKHHUBUHAYHZVAYHLSFHUVZHIHHZPLZAHIHKLCLYKHKKLZWPLYAVVZPZLN
BPHZVHUHUKVJVUSVZVQVZHIPLYAVZ

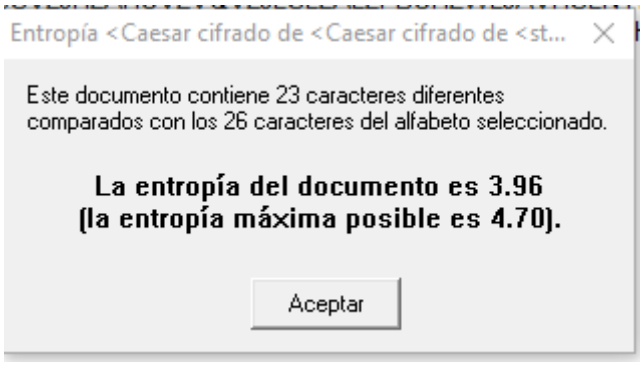
Cryptool nos ofrece varias herramientas que nos ayudan a la hora de descifrar textos cifrados de sustitución monoalfabeto.

En el apartado de análisis, podemos encontrar primero, una serie de herramientas para el análisis, entre las que se encuentran, Entropía, Histograma y N-grama.



ENTROPÍA:

Nos indica la cantidad de caracteres diferentes que tiene el texto, esto puede servir para ver si es válido aplicar análisis de frecuencias o no.

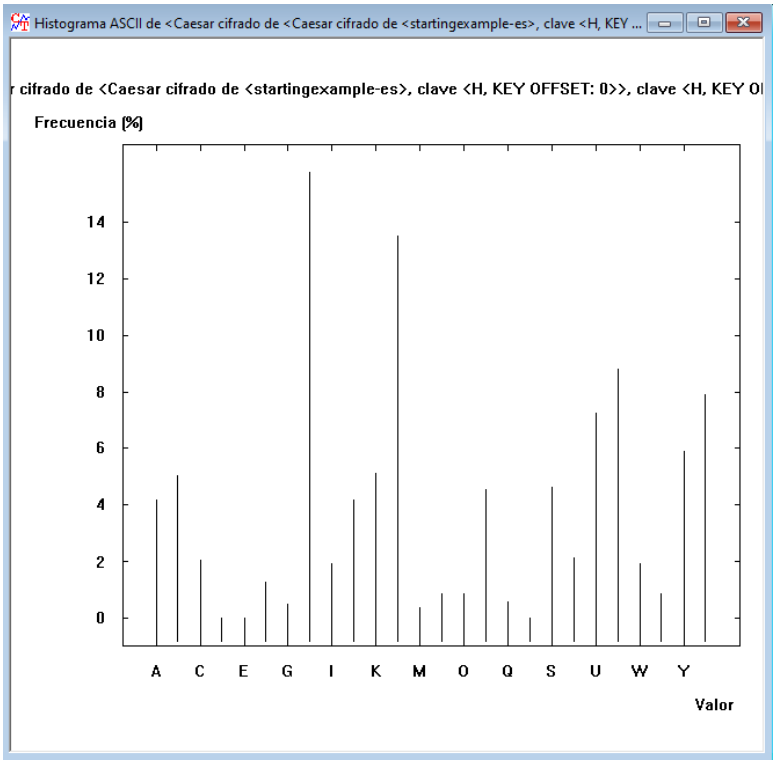


HISTOGRAMA:

Nos muestra un gráfico con la frecuencia en la que aparecen cada uno de los caracteres del texto cifrado, para poder así hacer un estudio y poder concluir la clave con la que se cifró.

En este gráfico se puede observar que los dos caracteres con más frecuencia son la H y la L,

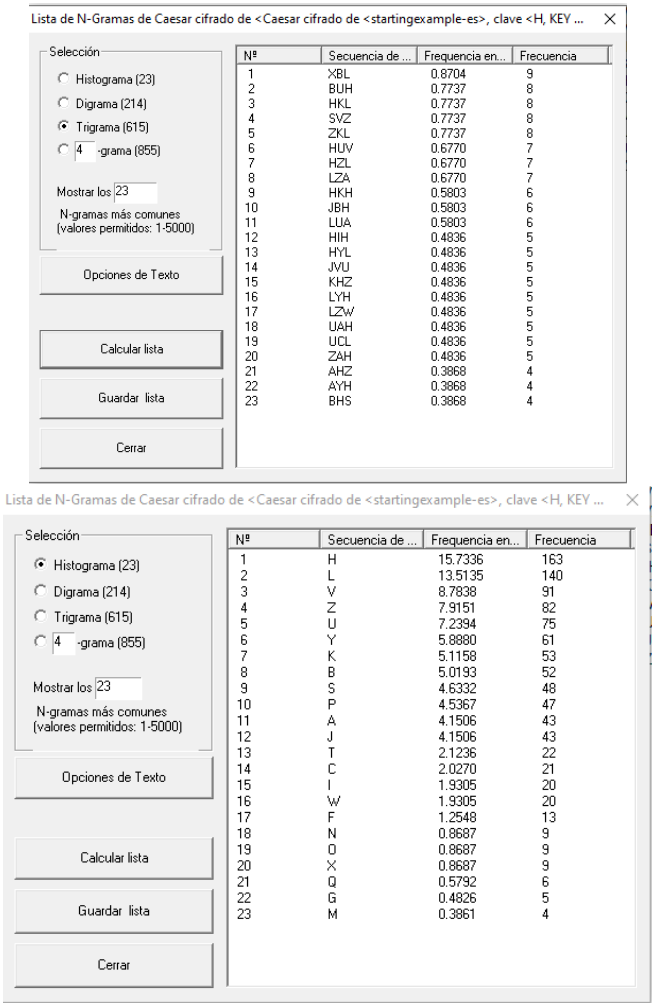
Podríamos así concluir que la frecuencia de H es la frecuencia de A en el alfabeto español y el de la L sería el de la E.



N-GRAMA:

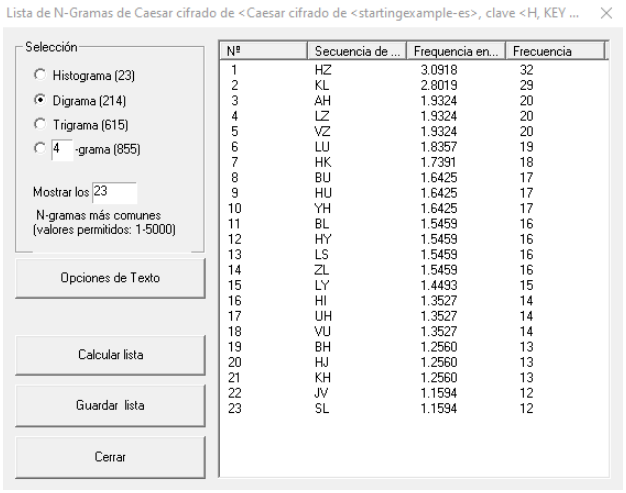
Nos da una amplia selección de n-gramas, entre ellos histogramas, diagrama, trigrama, y del tamaño que nosotros le indiquemos, es decir, todas las posibilidades.

En cuanto al Histograma, nos muestra la secuencia de 1 sola letra, lo que sería lo mismo que las frecuencias mostradas en el gráfico anterior por el histograma.

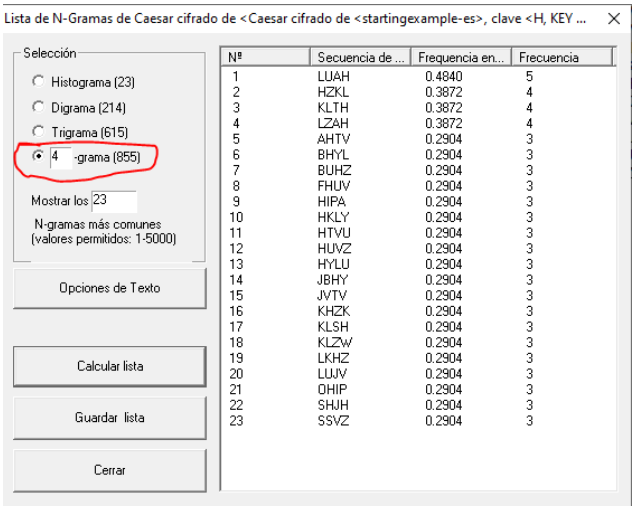


En cuanto al diagrama y el trigrama ocurre lo mismo, muestra la secuencia de 2 y 3 caracteres ordenados por mayor a menos frecuencia.

Esto nos puede ayudar a encontrar conjuntos de caracteres que nos ayuden a descifrar la clave, el caso más claro es en el inglés con los caracteres T y H, que aparecen con mucha frecuencia en los textos.



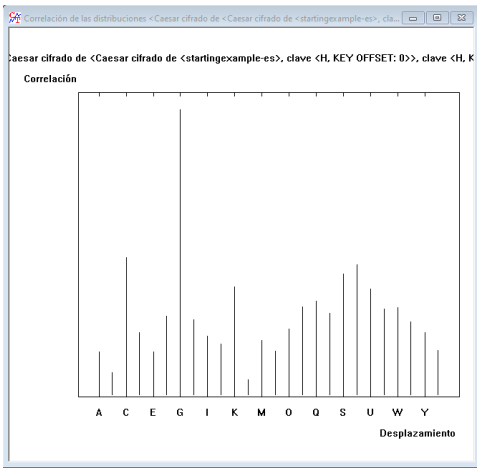
Y ocurre también lo mismo con los n-gramas más grandes, podemos seleccionar nosotros la longitud de estos. Estas se pueden guardar en listas.



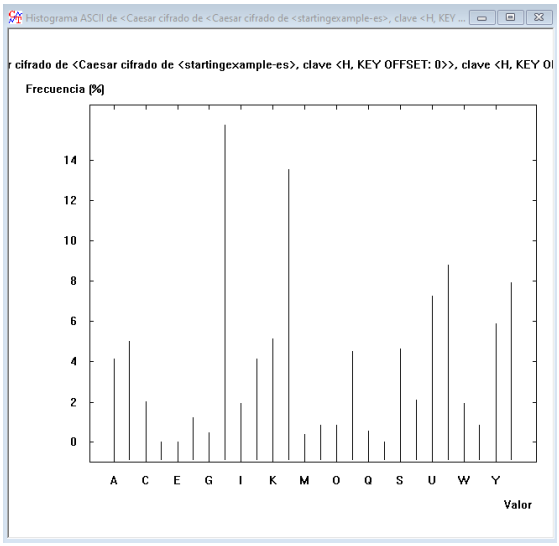
<u>ANALISIS DE CÉSAR:</u>

Primero nos pregunta por una posible clave que *Cryptool* calcula.

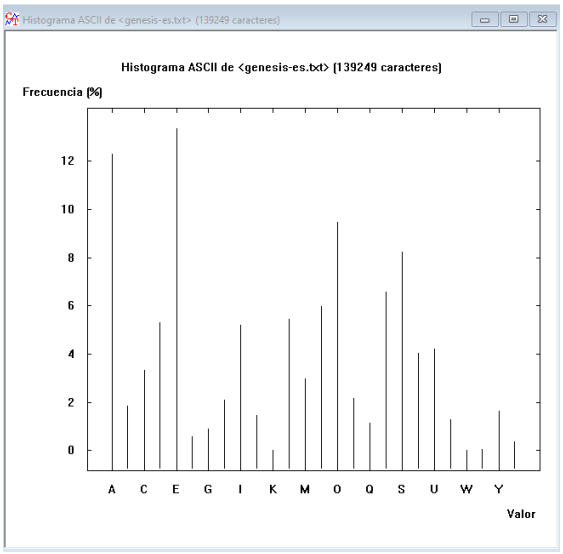
Nos ofrece una serie de información, el <u>desplazamiento:</u> la posibilidad de que sea el desplazamiento



Nos muestra un gráfico con las frecuencias de los caracteres que más aparecen:



Nos muestran el histograma con las frecuencias reales de los caracteres en el alfabeto español, para poder así comparar con el histograma anterior.

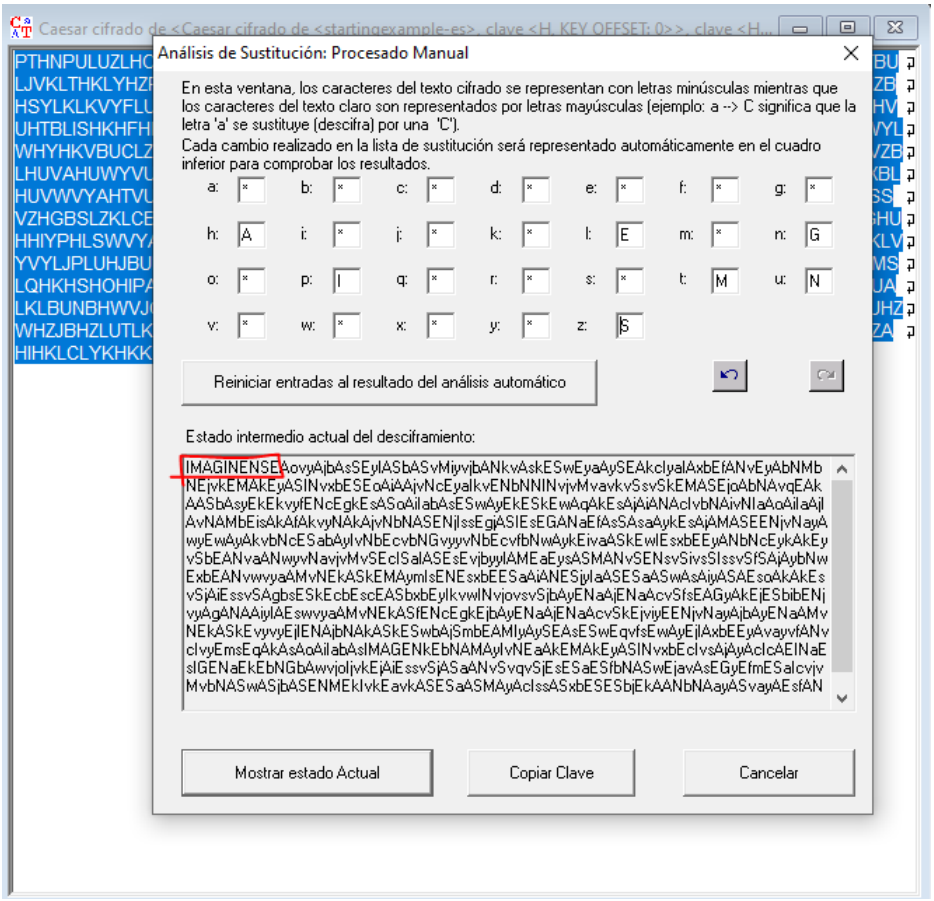


PROCESADO MANUAL:

En este, vamos introduciendo nosotros a mano los caracteres por lo que sustituimos cada una de las letras del alfabeto, para esto podemos utilizar las técnicas anteriormente mencionadas y mediante el análisis de frecuencia intentar averiguar cuál es la clave y poder así sustituir cada letra por su letra correspondiente.

Basándonos en el ejemplo dado y por el análisis anteriormente hecho, sustituimos con un desplazamiento 7.

Sustituyendo las letras necesarias para la primera palabra (IMAGINENSE)



Gracias a la ayuda proporcionada por todas estas herramientas de Cryptool podemos descifrar los textos encriptados con cifrados monoalfabeto con mayor facilidad.

Siguiendo el ejemplo dado, el texto final descifrado quedaría como el siguiente:

IMAGINENSEAHORACUALSERIASUASOMBROCUANDOALDESPERTARSEADVIRTIAQUEYANOERAUNMU
NECODEMADERASINOQUESEHABAACONVERTIDOENUNNINOCOMOTODOSLOSDEMASECHAUNAOJE
ADAASUALREDEDORYENVEZDELASHABITUALESPAREDESDEPAJADELACABANAVIOUNABONITAHABIT
ACIAONAMUEBLADAYADORNADACONUNASENCILLEZCASIELEGANTEYALSALTARDELACAMASEENCON
TRAPREPARADOUNVESTUARIONUEVOUNGORRONUEVOYUNPARDEBOTASDEPIELQUEERANUNVERD
ADEROSUEANOTANPRONTOCOMOSEVISTIASELEOCURRIAMETERLASMANOSENLOSBOLSILLOSYSACA
RUNPEQUEANOPORTAMONEDASDEMARFILENELQUEESTABANESCRITASESTASPALABRASAELHADADE
LOSCABELLOSAZULESDEVUELVEASUQUERIDOPINOCHOLOSCUARENTACENTAVOSYLEAGRADECESUB
UENCORAZANAABRIAELPORTAMONEDASYENVEZDECUARENTACENTAVOSDECOBREENCONTRACUAR
ENTAMONEDASDEORORECIENACUNADASDESPUACSFUEAMIRARSEALESPEJOYLEPARECIAQUEERAOT
ROYANOVIOREFLEJADALAHABITUALIMAGENDEUNAMARIONETADEMADERASINOQUEVIOLACARAVIV
AEINTELIGENTEDEUNGUAPOCHICODECABELLOSCASTANOSOJOSCELESTESYUNASPECTOALEGREYFEST
IVOCOMOUNASPASCUASENMEDIODETODASESTASMARAVILLASQUESESUCEDAANUNATRASOTRAELY
ANOSABAASIESTABADEVERDADDESPIERTOOSISEGUIASOANANDOCONLOSOJOSABIERTOS

2) Seleccionen tres fragmentos de texto con las siguientes características:

- Idioma español
- Con sentido (idealmente, el fragmento de un libro, mensaje, etc.)
- Entre 1000 y 1500 caracteres
- El texto sólo puede contener caracteres del alfabeto mostrado en la Tabla 1. Si en el texto original hay otros caracteres, deben ser sustituidos o eliminados. Por ejemplo, una 'Ñ', puede ser reemplazada por 'N', 'NH' o 'NN'. Deben eliminarse las tildes, diéresis, espacios y signos de puntuación.

a) Guarden estos fragmentos como GN_fragmento1.txt, GN_fragmento2.txt y GN_fragmento3.txt.

Los tres fragmentos pedidos están en el archivo comprimido como "5.1.4_fragmento1.txt", "5.1.4_fragmento2.txt" y "5.1.4_fragmento3.txt" respectivamente.

3) Ideen e implementen en Python un algoritmo de cifrado **simétrico** por **sustitución monoalfabeto**.

a) Expliquen el mecanismo de funcionamiento del algoritmo.

- El algoritmo únicamente debe aceptar como entrada caracteres del alfabeto mostrado en la Tabla 1. Esos son también los únicos caracteres válidos para la salida.

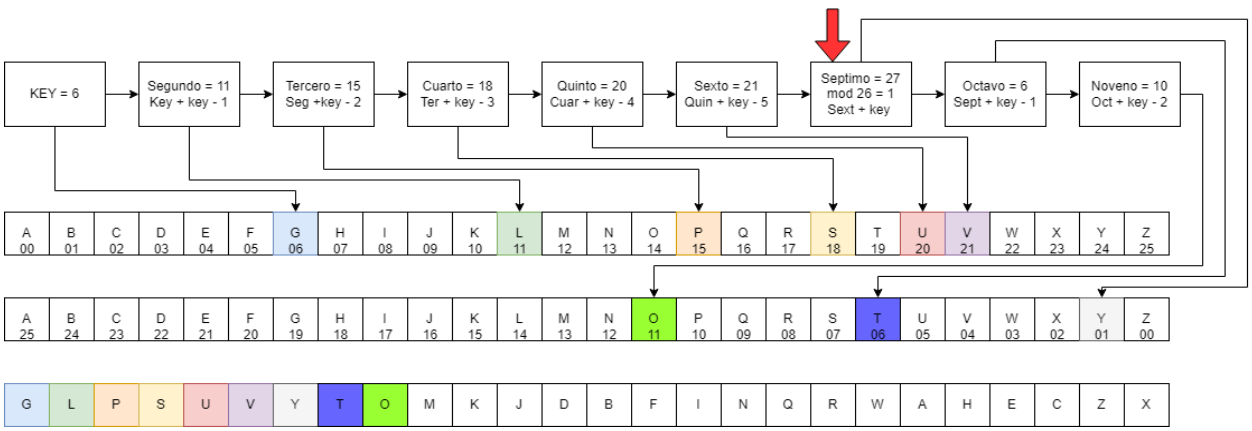
El algoritmo de cifrado monoalfabeto que hemos diseñado se basa en operaciones de salto y modulares sobre el alfabeto. El usuario que desea cifrar el texto proporciona una clave, en este caso numérica, y sobre esta clave se realizan unas operaciones:

- Se divide la clave entre 26, si el resultado es mayor que uno, el cifrado se repetirá ese número de veces, cada vez basándose en el alfabeto cifrado previamente.
- La clave se divide entre 26 y el resto (clave mod 26) se utiliza para cifrar la primera letra del alfabeto.
- El módulo se incrementa en módulo-1 para cifrar la segunda letra del alfabeto, módulo 26 - 2 para la tercera letra...
- Cuando llega a 0, el cifrado se continúa con el módulo original y se repite.

La base del cifrado resulta en seleccionar la primera letra del alfabeto original y cifrarla como la letra número clave mod 26 del alfabeto e ir realizando saltos cada vez más pequeños para disminuir las probabilidades de que se repitan los alfabetos cifrados. Para conseguir que se repitan menos los alfabetos también hemos implementado, que cuando se alcanza el final del alfabeto, en vez de volver al inicio de este e ir seleccionando las letras que todavía no están, se continúe como un ciclo desde el final hacia el principio, como un rebote. Cuando el algoritmo selecciona una letra que ya aparece en el algoritmo de cifrado, no la repite si no que selecciona la siguiente que no esté dentro del alfabeto cifrado en el orden que le corresponde, tanto hacia delante como inverso.

Vamos a ejemplificar toda esta explicación para que quede un poco más claro. En la figura que mostramos a continuación se puede observar cómo se realiza una parte de la primera iteración de nuestro algoritmo. Para este ejemplo hemos seleccionado la clave de cifrado 6, ya que es lo suficientemente baja como para mostrar el cuarto punto explicado anteriormente de que llegue a 0 y es lo suficientemente alta como para mostrar una inversión de alfabeto al llegar al final de este.

Se aprecia en la imagen como el tamaño del salto va decrementando progresivamente hasta llegar a 1, entre la “V” y la “W”, donde volvemos a tener un salto del tamaño de la clave y coincide con la primera inversión del algoritmo, marcada por la flecha roja en el séptimo paso



Una peculiaridad de nuestro algoritmo es que, al estar basado en operaciones modulares, todas las claves que coinciden con el 0 en módulo 26 (0, 26, 52, ...) dan como resultado de cifrado el propio alfabeto original, por lo tanto, todas estas claves deberían no ser utilizadas como ya se explica en la página de ayuda del código.

Además del código que se pide en el siguiente apartado, también incluimos un documento "keys.txt", que contiene todas las claves desde el número 1 hasta el 2599, mostradas en el siguiente formato:

Número de clave [ALFABETO CIFRADO.....] Número de clave en módulo 26

En este documento se puede comprobar con facilidad como cada 0 en módulo 26 tiene la misma clave y también se puede comprobar como en estas 2600 primeras claves ya hay algunas que figuran repetidas.

b) Guarden el código implementado en un archivo denominado monoalfabeto.py.(o monoalfabeto.zip, en caso de que se utilice más de un archivo).

- El algoritmo debe poder ejecutarse desde una terminal, invocando al comando monoalfabeto.py

Adjuntamos un comprimido "monoalfabeto.zip" con el código fuente "monoalfabeto.py", el documento de claves explicado previamente "keys.txt" y la figura previa completa como "monoalfabeto.png".

Respecto a la ejecución del algoritmo, a pesar de tener en la primera línea la cabecera *hash-bang* para ser ejecutado con Python, no estamos seguros de que funcione correctamente por lo que recomendamos la ejecución como python monoalfabeto.py.

c) Indiquen la sintaxis de uso del comando monoalfabeto.py.

A continuación, se muestra un extracto de la página de ayuda de nuestro código, donde se explican las diferentes opciones de las que dispone el programa.

Usage:

monoalfabeto.py [OPTIONS] INPUT_FILE KEY

monoalfabeto.py [OPTIONS] -i INPUT_FILE -k KEY

OPTIONS:

-i INPUT_FILE, --input=INPUT_FILE -> Set the name of the input file

-k KEY, --key=KEY -> Set the key to the cipher

-o OUTPUT_FILE, --output=OUTPUT_FILE -> Set the name of the output file, default monoalfabeto.mono

-d, --decipher -> Decipher the text instead of ciphering it

-h, --help -> Display manual page

Como explicación breve podemos decir que tiene dos modos básicos, el de cifrado, acompañando el nombre del programa con el fichero de entrada y la clave de cifrado y el de descifrado, igual que el anterior, pero con la opción -d. Opcionalmente podemos indicar el fichero de salida con la opción -o y también mostrar la página de ayuda con el *flag* –h.

4) Cifren, utilizando el código implementado en el ejercicio 3), el texto de los archivos `GN_fragmento1.txt` y `GN_fragmento2.txt`.

a) Indiquen en la memoria las instrucciones para descifrar los textos (comando y parámetros necesarios)

Según el código desarrollado en el ejercicio anterior y los fragmentos del ejercicio 2, para descifrar los fragmentos tendremos que ejecutar:

· python monoalfabeto.py -d –i 5.1.4_fragmento1.mono -k 22

· python monoalfabeto.py -d –i 5.1.4_fragmento2.mono -k 33

Opcionalmente en cualquiera de los dos se puede indicar el fichero al que se quieren sacar los textos descifrados con la opción "-o [NOMBRE_DE_FICHERO]".

b) Guarden los textos cifrados como `GN_fragmento1.mono` y `GN_fragmento2.mono` respectivamente y súbanlos a la carpeta `P1_Retos` en Teams.

Ambos fragmentos están subidos en la carpeta de Teams con los nombres 5.1.4_fragmento1.mono y 5.1.4_fragmento2.mono.

5) Implementen en Python el algoritmo de Vigenère.

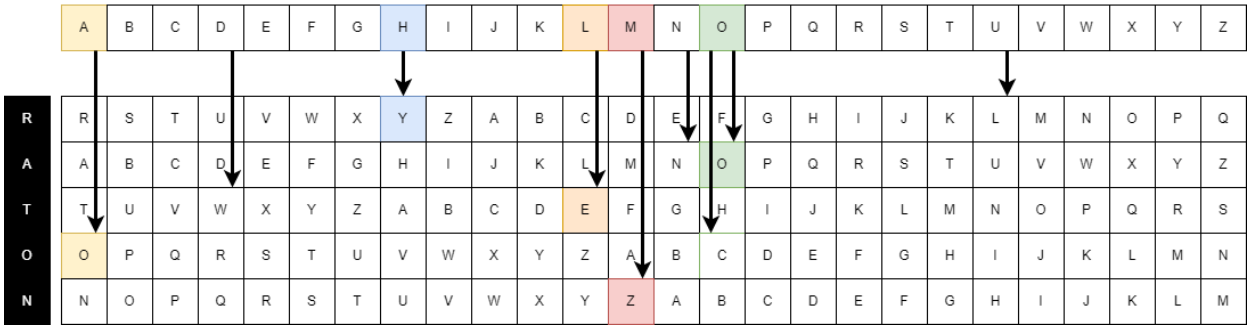
a) Expliquen el mecanismo de funcionamiento del algoritmo.

- El algoritmo únicamente debe aceptar como entrada caracteres del alfabeto mostrado en la Tabla 1. Esos son también los únicos caracteres válidos para la salida.

El algoritmo de Vigenère se basa en el cifrado monoalfabeto de un texto que previamente es dividido en función de la longitud de la clave, que también marca los desplazamientos de los alfabetos con los que se cifra cada una de las divisiones.

Si, por ejemplo, la clave de cifrado seleccionada es RATON, el texto se irá dividiendo en grupos de 5 letras, y cada una de ellas se cifra en función del cifrado César cuya primera letra es la que le corresponde de la clave. Para este ejemplo, tenemos un texto como HOLA MUNDO, que nos quedaría de la siguiente manera:

H	O	L	A	M	U	N	D	O
R	A	T	O	N	R	A	T	O
Y	O	E	O	Z	L	N	W	C



Como se puede apreciar en la imagen, cada letra de las cinco primeras se cifra en base a una clave César diferente, que además se repite cada cinco letras ya que esa es la longitud de nuestra clave.

En el código se realiza esta transformación utilizando los valores Unicode de cada una de las letras del texto sumada al código Unicode de la letra que le corresponde de la clave y el resultado se vuelve a pasar a texto. Estas dos operaciones se realizan con las funciones “ord()” y “chr()”.

b) Guarden el código implementado en un archivo `vigenere.py` (o `vigenere.zip`, en caso de que se utilice más de un archivo).

- El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `vigenere.py`

Se entrega un comprimido “vigenere.zip” que contiene el código completo del programa, “vigenere.py”, junto con la imagen previa de muestra “vigenere.png”

c) Indiquen la sintaxis de uso del comando `vigenere.py`.

Se adjunta la salida del comando de ayuda del programa:

Usage:

vigenere.py [OPTIONS] INPUT_FILE KEY

vigenere.py [OPTIONS] -i INPUT_FILE -k KEY

OPTIONS:

-i INPUT_FILE, --input=INPUT_FILE -> Set the name of the input file

-k KEY, --key=KEY -> Set the key to the cipher

-o OUTPUT_FILE, --output=OUTPUT_FILE -> Set the name of the output file, default vigenere.vig

-d, --decipher -> Decipher the text

-h, --help -> Display manual page

Como breve explicación, es muy similar al monoalfabeto.py, donde tenemos dos modos, cifrado y descifrado. Para cifrar tendremos que introducir el fichero a cifrar más la clave y para descifrar con añadir el *flag* –d será suficiente. De igual manera que en el monoalfabeto podemos fijar un fichero de salida con el *flag* -o [FICHERO_DE_SALIDA]. Tanto el fichero de entrada como al clave se pueden pasar en ese orden sin *flags* o con el *flag* –i –k respectivamente.

6) Cifren, utilizando el código implementado en el ejercicio 5) el texto del archivo `GN_fragmento3.txt`.

a) Indiquen en la memoria las instrucciones para descifrar los textos (comando y parámetros necesarios)

- La contraseña puede tener, como máximo, 7 caracteres

Según el código desarrollado en el ejercicio anterior y el fragmento del ejercicio 2, para descifrar el fragmento tendremos que ejecutar:

· python vigenere.py -i .\5.1.4_fragmento3.vig -k JAMAICA -o .\5.1.4_fragmento3.txt -d

Opcionalmente en cualquiera de los dos se puede indicar el fichero al que se quieren sacar los textos descifrados con la opción "-o [NOMBRE_DE_FICHERO]".

b) Guarden el texto cifrado como `GN_fragmento3.vig` y súbanlo a la carpeta `P1_Retos` en Teams.

El fragmento está subido en la carpeta de Teams con el nombre 5.1.4_fragmento3.vig.

7) Seleccionen dos fragmentos de tipo `*.mono` de otro grupo.

a) En la memoria se debe indicar el nombre del fragmento seleccionado. No es necesario incluir el fragmento seleccionado.

Los dos fragmentos escogidos para descifrar son los siguientes:

· 5.2.3_fragmento1.mono

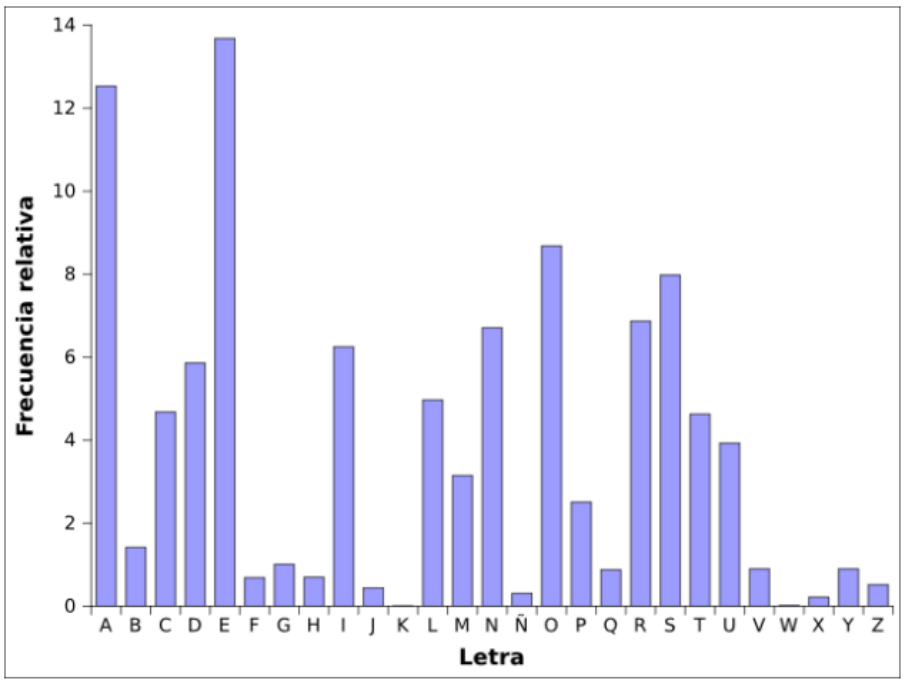
· 5.2.3_fragmento2.mono

b) Traten de obtener el texto en claro correspondiente, aplicando análisis de frecuencias. Indiquen los pasos que se han seguido.

Para realizar el análisis de frecuencias nos apoyaremos en *Cryptool* y en las herramientas de análisis que este nos proporciona sin realizarlo de manera automática. Primero copiamos el fragmento en *Cryptool* y luego sacamos los histogramas de las letras con sus frecuencias y también de los digramas con sus frecuencias. Con estos datos podemos realizar las primeras suposiciones.

Nº	Subcadena	Frequencia (en %)	Frecuencia
1	R	13.1561	188
2	J	12.5262	179
3	L	10.7068	153
4	W	7.9076	113
5	T	7.4878	107
6	S	6.3681	91
7	C	5.4584	78
8	X	5.4584	78
9	Z	4.8985	70
10	E	4.8286	69
11	N	3.7789	54
12	I	2.9391	42
13	H	2.6592	38
14	P	2.0994	30
15	A	1.8195	26
16	K	1.8195	26
17	Y	1.7495	25
18	F	1.1896	17
19	V	0.9797	14
20	O	0.9097	13
21	M	0.7698	11
22	G	0.3499	5
23	U	0.1400	2

Nº	Subcadena	Frequencia (en %)	Frecuencia
1	RW	2.6611	38
2	RT	2.4510	35
3	CR	2.3810	34
4	JW	2.3109	33
5	RS	2.2409	32
6	XR	2.2409	32
7	PX	2.1008	30
8	JT	2.0308	29
9	SJ	1.9608	28
10	CL	1.6106	23
11	SR	1.5406	22
12	LT	1.4706	21
13	TR	1.4706	21
14	LC	1.4006	20
15	EJ	1.3305	19
16	WI	1.3305	19
17	LW	1.1905	17
18	WL	1.1905	17
19	JS	1.1204	16
20	LE	1.1204	16
21	NL	1.1204	16
22	TJ	1.0504	15
23	WC	1.0504	15



Frecuencia de las letras (%) en español.

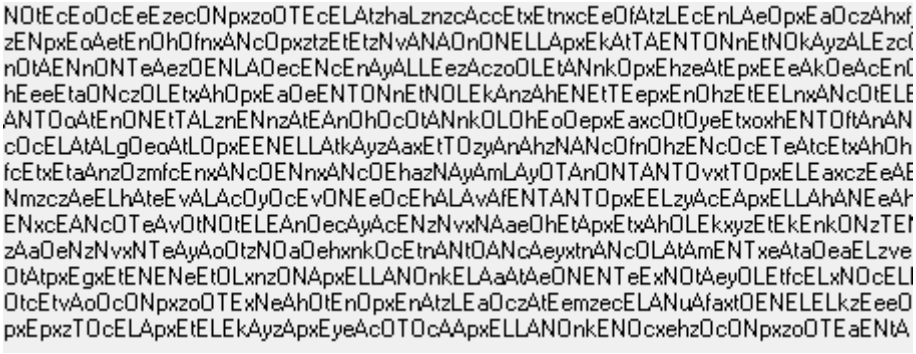
Como podemos ver en el gráfico de barras, las letras más comunes en español son la E y la A, con bastante diferencia con respecto al resto. Por lo tanto, podemos suponer que:

{R} → {E} y {J} → {A}

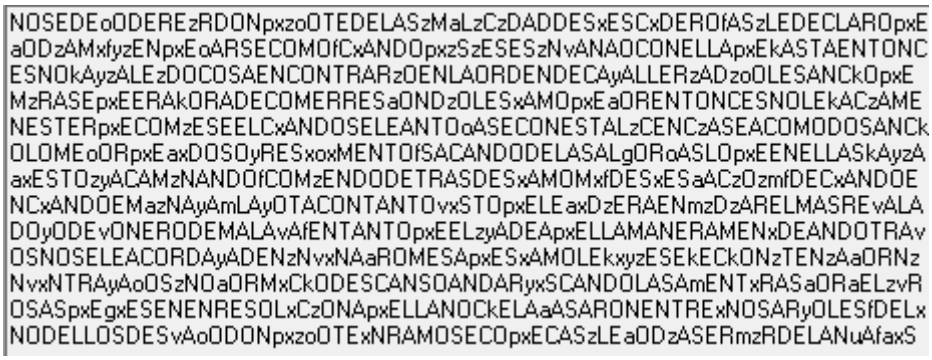
Esta suposición es además respaldada por la información de los digramas, donde los más frecuentes son RW y RT, que podrían ser ES y EL, que son digramas frecuentes en español. Pero de momento vamos a no suponer nada en base a los digramas y continuamos con las frecuencias individuales.

Las siguientes letras más comunes de la lenga española son {O,S,R,N,I,D} en ese orden, con unas frecuencias desde 8.6 hasta 5.8, como la {O} tiene una frecuencia relativamente muy superior al resto podemos inferir que {0} → {L}. Con esta nos queda un conjunto con cinco letras {S,R,N,I,D} para sustituir por las cinco siguientes según las frecuencias {W,T,S,C,X}. Dentro de este grupo, para obtener las coincidencias iremos sustituyendo en el texto, apoyándonos en *Cryptool* y el análisis manual para no tener que realizar las sustituciones de una forma manual. Al ir realizando las sustituciones con este conjunto y sus posibles variaciones puede parecer que la {N} corresponde a la {W} ya que encontramos en varias partes del texto la sílaba NO, que puede formar parte de otras palabras o ser una en sí misma, por lo que aceptaremos esta suposición. De igual manera, al sustituir la {I} por la {T} nos aparece en varias ocasiones la sílaba TE, por lo que también aceptamos esta suposición. Si añadimos la {L} al conjunto y probamos sustituciones, vemos que con las que ya tenemos y la {S} por la {L} nos aparece en varias ocasiones la sílaba LE y en alguna ocasión la palabra ELLA.

Con estas letras ya tenemos una gran parte del texto descifrado y podemos empezar a intuir letras a partir de las que ya tenemos o las que faltan. En la siguiente imagen vemos un trozo del fragmento con las sustituciones realizadas hasta el momento.



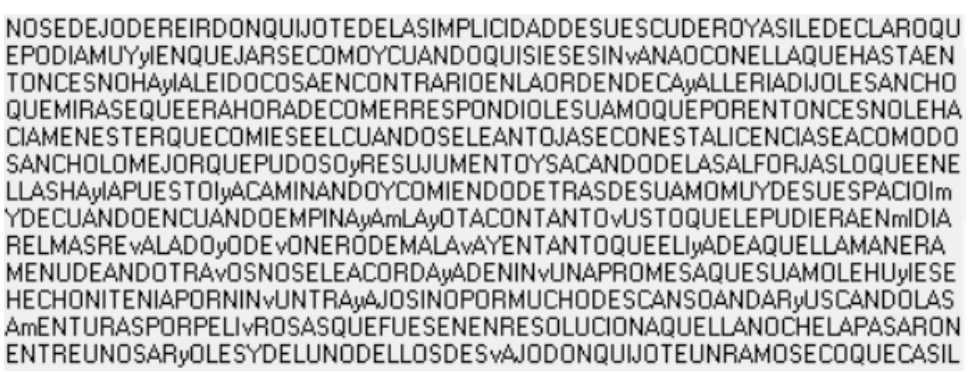
Apoyándonos en las frecuencias del lenguaje y en el contexto del fragmento, empezamos a inferir más sustituciones, como la {N} por la {C}, como se ve en la segunda línea donde al realizar esta sustitución pone CON ELLA. Del mismo modo, aparece la {E} en un fragmento del tipo “CONTeA”, por lo que puede ser la {R}. Así continuamos entre las frecuencias y las letras restantes del texto para ir obteniendo todas las sustituciones y hallar la clave. Las próximas sustituciones que obtenemos son las siguientes {H} → {M}, {T} → {S}, {C} → {D}. Con estas sustituciones ya tenemos bastante texto en claro y legible como podemos ver en la imagen a continuación.



Aquí las sustituciones restantes casi se pueden realizar todas por el contexto, vemos que la {X} corresponde a la {U} y nos quedan muchas {P} detrás de la ya sustituida {U}, por lo que será la {Q}.

La {K} corresponde a la {H} dejándonos ver palabras como HASTA. En la primera línea para “despejar ” la palabra QUIJOTE, realizamos las sustituciones {O} → {J} y {Z} → {I}. Sustituyendo la {A} por la {P} nos aparece la palabra SIMPLICIDAD en la primera línea, con la {G} por la {F} nos aparecen ALFORJAS, la {F} es la {Y} ya que aparece sola entre dos palabras y después de MU, formando MUY.

Nos quedan así siete sustituciones por realizar, de las cuales en el texto figuran cuatro, {Y,M,V,U}, estas por contexto las sacamos sin mayor dificultad, ya que como se ve en la imagen quedan pocas palabras por descifrar.



La {Y} corresponde a la {B}, la {M} a la {V}, la {V} a la {G} y la {U} a la {Z}. Nos quedan tres sustituciones sin realizar en la clave, {K,W,X}, no sabemos que letras son cifradas de las siguientes tres {B,D,Q}.

Para el segundo texto realizamos la misma operación con unos pasos similares.

Nº	Subcadena	Frequencia (en %)	Frecuencia
1	G	13.7308	152
2	Q	11.2918	125
3	U	9.1238	101
4	K	8.2204	91
5	W	6.9557	77
6	T	6.5944	73
7	Y	5.9621	66
8	D	5.7814	64
9	P	4.8780	54
10	V	4.7877	53
11	B	4.6974	52
12	S	4.6070	51
13	M	3.4327	38
14	L	3.0714	34
15	X	1.8067	20
16	H	1.0840	12
17	O	0.9033	10
18	Z	0.7227	8
19	C	0.6323	7
20	N	0.6323	7
21	F	0.4517	5
22	I	0.3613	4
23	J	0.1807	2
24	R	0.0903	1

Nº	Subcadena	Frequencia (en %)	Frecuencia
1	EN	9.8039	10
2	DE	5.8824	6
3	FR	3.9216	4
4	CI	2.9412	3
5	IA	2.9412	3
6	IS	2.9412	3
7	NC	2.9412	3
8	NT	2.9412	3
9	RE	2.9412	3
10	TE	2.9412	3
11	UE	2.9412	3
12	AD	1.9608	2
13	AM	1.9608	2
14	CU	1.9608	2
15	EC	1.9608	2
16	ES	1.9608	2
17	HI	1.9608	2
18	MA	1.9608	2
19	ME	1.9608	2
20	NA	1.9608	2
21	OR	1.9608	2
22	RA	1.9608	2
23	TO	1.9608	2
24	AG	0.9804	1
25	AN	0.9804	1
26	AR	0.9804	1

En este caso no vamos a realizar un paso a paso tan detallado como en el caso anterior porque el procedimiento es el mismo. Comenzamos con las dos letras con mayor probabilidad y las posibilidades de las 4-5-6 letras siguientes y los digramas para obtener las primeras silabas en el texto, a partir de estas vamos sacando las que son posibles por contexto y cuando no tenemos esa posibilidad nos apoyamos en las frecuencias de cada una de las letras para probar suposiciones. La parte más importante para nosotros en este cifrado es que vuelve a haber dos letras que no aparecen en el texto que no podemos identificar a cuál pertenecen, en este caso nos referimos al conjunto {A,E} que descifradas son {K,W}, pero no sabemos cuál es cuál.

c) Muestren el texto descifrado.

El primer texto descifrado y espaciado es como sigue:

NO SE DEJO DE REIR DON QUIJOTE DE LA SIMPLICIDAD DE SU ESCUDERO Y ASI LE DECLARO QUE PODIA MUY BIEN QUEJARSE COMO Y CUANDO QUISIESE SIN GANA O CON ELLA QUE HASTA ENTONCES NO HABIA LEIDO COSA EN CONTRARIO EN LA ORDEN DE CABALLERIA DIJOLE SANCHO QUE MIRASE QUE ERA HORA DE COMER RESPONDIOLE SU AMO QUE POR ENTONCES NO LE HACIA MENESTER QUE COMIESE EL CUANDO SE LE ANTOJASE CON ESTA LICENCIA SE ACOMODO SANCHO LO MEJOR QUE PUDO SOBRE SU JUMENTO Y SACANDO DE LAS ALFORJAS LO QUE EN ELLAS HABIA PUESTO IBA CAMINANDO Y COMIENDO DETRAS DE SU AMO MUY DE SU ESPACIO IV Y DE CUANDO EN CUANDO EMPINABA V LA BOTA CON TANTO GUSTO QUE LE PUDIERA ENVIDIAR EL MAS REGALADO BODEGONERO DE MALAGA Y EN TANTO QUE EL IBA DE AQUELLA MANERA MENUDEANDO TRAGOS NO SE LE ACORDABA DE NINGUNA PROMESA QUE SU AMO LE HUBIESE HECHO NI TENIA POR NINGUN TRABAJO SINO POR MUCHO DESCANSO ANDAR BUSCANDO LAS AVENTURAS POR PELIGROSAS QUE FUESEN EN RESOLUCION AQUELLA NOCHE LA PASARON ENTRE

UNOS ARBOLES Y DE L UNO DE LLOS DESGAJO DON QUIJOTE UN RAMO SECO QUE CASI LE PODIA SERVIR DE LANZA Y PUSO EN EL EL HIERRO QUE QUITO DE LA QUE SE LE HABIA QUEBRADO TODA AQUELLA NOCHE NO DURMIO DON QUIJOTE PENSANDO EN SU SENHORA DULCINEA POR ACOMODARSE A LO QUE HABIA LEIDO EN SUS LIBROS CUANDO LOS CABALLEROS PASABAN SIN DORMIR MUCHAS NOCHES EN LAS FLORESTAS Y DESPOBLADOS ENTRETENIDOS CON LAS MEMORIAS DE SUS SENHORAS NO LA PASO ANSI SANCHO PANZA QUE COMO TENIA EL ESTOMAGO LLENO Y NO DE AGUA DE CHICORIA DE UN SUENHO SE LA LLEVO TODA Y NO FUERAN PARTE PARA DESPERTARLE SI SU AMO NO LO VI LLAMAR A LOS RAYOS DEL SOL QUE LE DABAN EN EL ROSTRO NI EL CANTO DE LAS AVES QUE MUCHAS Y MUY REGOCIJADAMENTE LA VENIDA DEL NUEVO DIA SALUDABAN AL LEVANTARSE DIO UN TIENTO A LA BOTA Y HALLOLA ALGO MAS FLACA QUE LA NOCHE ANTES

El segundo texto descifrado y espaciado es como sigue:

EN CRIPTOGRAFIA EL CIFRADO CESAR TAMBIEN CONOCIDO COMO CIFRADO POR DESPLAZAMIENTO CODIGO DE CESAR O DESPLAZAMIENTO DE CESAR ES UNA DE LAS TECNICAS DE CIFRADO MAS SIMPLES Y MAS USADAS ES UN TIPO DE CIFRADO POR SUSTITUCION EN EL QUE UNA LETRA EN EL TEXTO ORIGINAL ES REEMPLAZADA POR OTRA LETRA QUE SE ENCUENTRA UN NUMERO FIJO DE POSICIONES MAS ADELANTE EN EL ALFABETO POR EJEMPLO CON UN DESPLAZAMIENTO DE LA A SERIA SUSTITUIDA POR LA D SITUADA LUGARES A LA DERECHA DE LA A LA B SERIA REEMPLAZADA POR LA E ETC ESTE METODO DEBE SU NOMBRE A JULIO CESAR QUE LO USABA PARA COMUNICARSE CON SUS GENERALES EL CIFRADO CESAR MUCHAS VECES PUEDE FORMAR PARTE DE SISTEMAS MAS COMPLEJOS DE CODIFICACION COMO EL CIFRADO VIGENERE E INCLUSO TIENE APLICACION EN EL SISTEMA ROT COMO TODOS LOS CIFRADOS DE SUSTITUCION ALFABETICA SIMPLE EL CIFRADO CESAR SE DESCIFRA CON FACILIDAD Y EN LA PRACTICA NO OFRECE MUCHA SEGURIDAD EN LA COMUNICACION EL CIFRADO CESAR RECIBE SU NOMBRE EN HONOR A JULIO CESAR QUE SEGUN SU ETONIO LO USO CON UN DESPLAZAMIENTO DE TRES ESPACIOS PARA PROTEGER SUS MENSAJES IMPORTANTES DE CONTENIDO MILITAR AUNQUE CESAR ES LA PRIMERA PERSONA DE LA QUE SE SABE QUE HAYA USADO ESTE SISTEMA ANTERIORMENTE YA SE UTILIZARON OTROS CIFRADOS POR SUSTITUCION EL SOBRINO DE JULIO CESAR AUGUSTO TAMBIEN EMPLEO EL CIFRADO PERO CON UN DESPLAZAMIENTO DE UNO

d) Indiquen la clave de cifrado.

La clave del primer texto cifrado es <JYNCRGVKZO-SHWLAPETIXM--FU> donde los guiones representan que las letras no aparecen en el texto cifrado.

La clave del segundo texto cifrado es <QHYPGXOFWN-VMDULCTKBSJ-RIZ> donde los guiones representan que las letras no aparecen en el texto cifrado.

e) Expliquen, si es posible, el funcionamiento del algoritmo de cifrado analizado.

En cuanto al funcionamiento del algoritmo que ha cifrado estos textos, vamos a comparar cada uno de los cifrados con el alfabeto.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
J	Y	N	C	R	G	V	K	Z	O		S	H	W	L	A	P	E	T	I	X	M			F	U
Q	H	Y	P	G	X	O	F	W	N		V	M	D	U	L	C	T	K	B	S	J		R	I	Z

Realmente no hemos podido encontrar ningún patrón que relacione el primero con el segundo alfabeto y el original, por lo que no sabemos cómo funciona el algoritmo de cifrado.

8) Seleccionen un fragmento de tipo `*.vig` de otro grupo.

a) En la memoria se debe indicar el nombre del fragmento seleccionado. No es necesario incluir el fragmento seleccionado.

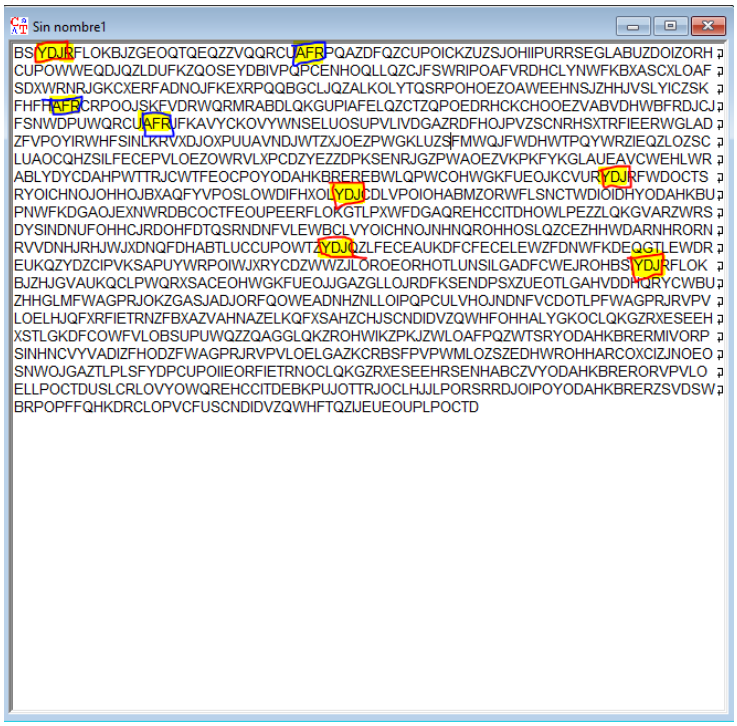
Nuestro fragmento seleccionado es el 5.2.3_fragmento3.vig

b) Traten de obtener el texto en claro correspondiente, aplicando Kasiski. Indiquen los pasos que se han seguido.

CRYPTOANÁLISIS DE VIGENÉRE CON EL METODO DE KASISKI:

Lo primero que debemos hacer es buscar cadenas de caracteres de al menos 3 letras que se repitan en el texto encriptado.

Podemos para ello contar las distancias que hay entre un grupo determinado de caracteres, por ejemplo, si cogemos el segundo conjunto de 3 letras (YDJ) en la posición = 3, vemos que se repite dentro de 565 caracteres, en 45, en 215 y se repite por última vez 105 caracteres más adelante.



Lo hacemos también con más conjuntos de letras de 3 o más caracteres que se repitan en el texto.

En el ejemplo de arriba también podemos ver el conjunto de (AFR)

Ahora, una vez que obtenemos las distancias de un número relativo de conjuntos de letras, pasamos a realizar el máximo común divisor de las distancias entre los que se encuentran los conjuntos de letras que se repiten.

En el caso de YDJ, sería el mcd(565, 45, 215, 105) = 5

En el caso de AFR, sería el mcd(195,80) = 5

En el caso de POI, sería mcd(577,268,435) = 1 (Este no nos sirve)

Si realizamos esto al resto de conjuntos de letras obtendremos algo parecido a lo siguiente.

Length	Count	Word	Factor	Location (distance)
3	5	YDJ	5	3 568 (565) 613 (45) 828 (215) 933 (105)
3	3	DJR	5	4 569 (565) 934 (365)
3	3	JRF	5	5 570 (565) 935 (365)
3	3	RFL	15	6 681 (675) 936 (255)
3	3	FLO	15	7 682 (675) 937 (255)
3	3	LOK	15	8 683 (675) 938 (255)
3	3	AFR	5	30 225 (195) 305 (80)
3	4	QZC	5	39 114 (75) 259 (145) 779 (520)
3	4	CUP	5	41 76 (35) 821 (745) 1321 (500)
3	4	UPO	5	42 77 (35) 822 (745) 1322 (500)
3	4	POI	1	43 620 (577) 888 (268) 1323 (435)
3	3	OIC	3	44 581 (537) 761 (180)
3	3	GLA	1	63 361 (298) 496 (135)
3	3	JQZ	5	85 180 (95) 830 (650)
3	3	QZL	5	86 431 (345) 831 (400)
3	3	LQZ	5	113 258 (145) 778 (520)
3	3	OAF	1	124 145 (21) 1205 (1060)
3	3	NWF	40	134 654 (520) 854 (200)
3	3	WFK	40	135 655 (520) 855 (200)
3	3	BXA	5	138 593 (455) 1103 (510)
3	3	NOJ	60	165 585 (420) 765 (180)
3	4	RPO	1	192 229 (37) 887 (658) 1452 (565)
3	5	OEZ	5	196 276 (80) 401 (125) 456 (55) 486 (30)
3	3	EEH	5	202 1157 (955) 1347 (190)
3	3	ZHH	115	208 783 (575) 1013 (230)

Debemos elegir entonces aquel factor que más se repita:

{5: 111, 15: 8, 3: 3, 40: 3, 60: 1, 115: 1, 10: 5, 4: 1}

Obtenemos así, el tamaño de la clave, en este caso sería 5.

Por lo cual, habrá 5 subcriptogramas que están cifrados con una misma letra, es decir, todas ellas serán cifras monoalfabéticas.

Dividimos el criptograma en tantos subcriptogramas como sea la longitud de la clave, en este caso 5.

Como cada uno de esos subcriptogramas es el resultado de una cifra monoalfabética con un desplazamiento, podemos aplicar análisis de frecuencia

BJKEEQAAZIZIRAOHWJUOBPOZWAHWAOXJENEQJKQHANHYKAPKWAKAZPHOAWJNWAKKNOLAHZHF WZYFKJUJJWZQHYEZAZELWPEKJAKKEEAYPJEYKEPWEVJOYNHAPWXJPARNIYKWANCEEKXACHEKZYNH OQNWYNNHZHNNNJNACWJEKEWWQWKZKYIYWRHNAEBJKJKWAWEAOKEAHWHWJAJONLPHNOWJL JFNANKAJIWHKKEXKWBWAKWKOZYKMPNAHWJLABPZHHXNNALPIFNKERAYKOLPULWCEJJJRJYKZWP KOUIWZEP

SRBOQQFZCCSISBICWQFSICQCRFCFSAWGROXBQOSOWSJIFFOFQBGFCOCOBBCWQFAOSSIZOSSIGFISR OAWOGSJWWQSOSCORCZSGOPGAHBCWCOOBBWGOURCOOOQODOCOBWCOOBFOWOORGWQCOZ GWSUCHSFBOOQOCWHRHWQBCTQCDCZFGDQCSWWCZOOSDJSRBGQQCGOZJSSOHQBGAOSOWHOC OFTAROQIZZAQHSDHAOGSSDFSQGZIJAWOBISCDOAROZSWSWACOWZSCIIOGSSBOBROOSOQCBOOJS OOBSBFDPSDHIOO

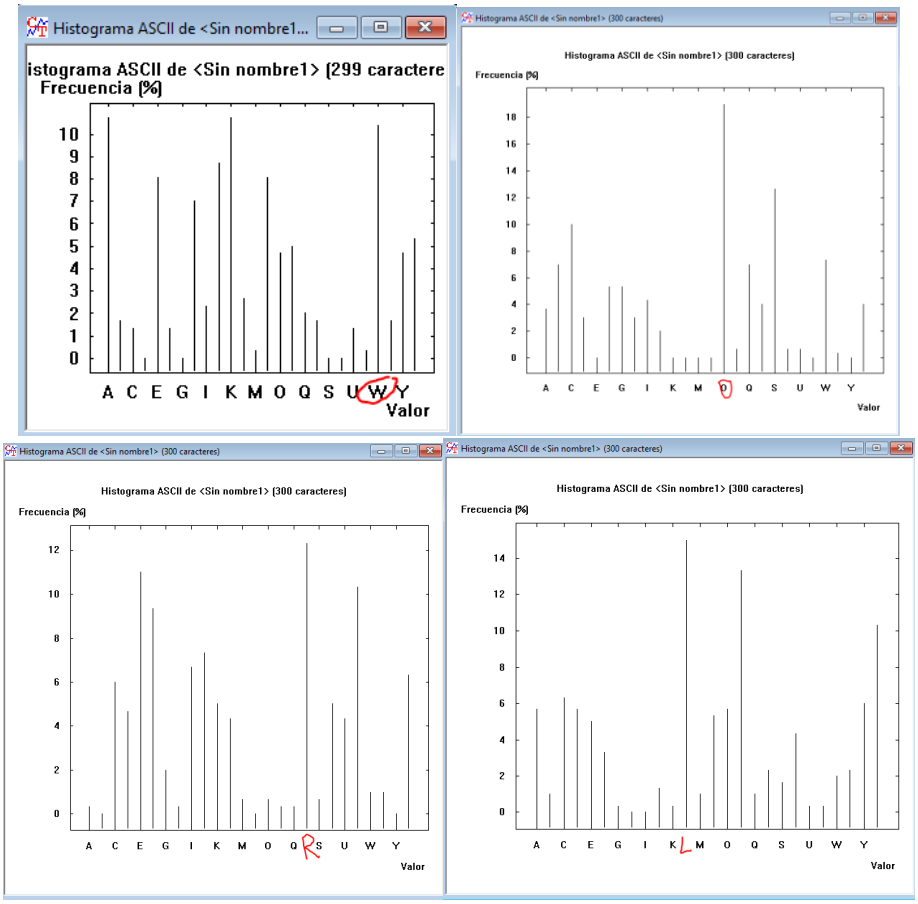
IFJQZRRDUKJPEUZUEZKEVELJIVLKCFRKFJRGZLREEJVCHROVRDUETEKEVFJDRRVVEUVRJCXELVRIVXVTE
KFFTRZCCIEEVDZEZEKLVLLDTWCDRWCKJRFTIJJFSILDIMFTIDUKJRCUFTFRIWZVRIFJFRVCIJRSEDRVJJFT
UZZEFEFKTRZIARJDJETIFRIFJVCREKJGREXTVRULGKJREZIUJVLGVEFEFVZFZCVFLCZETFVUZGRKZFTDRVI
VIDGVEKFMZRRIEOTFUEECZEECDRVECLVRIKTCLRIDRVRFRVCVFJUC

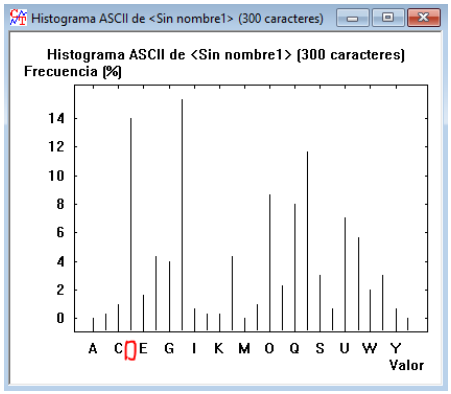
YLZTZCPFPZOUGZOPQLZYPNLFPRYBXSNCAFPCAYPZEZSZFCJDMLPLZDCZDRFPCJYYLPDDPNTEAPWNXP
NZZLMWPZLLQLPZLZDNPZFACWYATTPAELOFKYWSCOBYLFYLOZLWDAPDEDTPLLDETLLASNORDNLLCN
OLZAOVRXDLPYLACLDDLEYPPPXZLOLLCOYLZALXOFJLDNZLDYZMPZAFANPLNCPPPLXTBAEXCNZOYLRE
LCLPZLOZWPSAEONYZZPPLCPLEOCZOJLYPOTLRENZAEPLTCYETPTLPRPAEDPQCCNZTEPT

DOGQVUQQOUHRLDRODDQDQHQSODNXLDRXDKQLLTOOHHLSHRSRRQIQQRHVHDSUUFCWUVGFVR
RRDOHLDUDXPUWDQIOUHFVOXYPRWVYUWRDHRFOHRQHUCDDRHHXVOHDVHOSDHHNGXBFEOPG
HDPQRDDHDTDEVHHHQHRRDHDHUODFUFENEEUDVUORWORUGWHDOHUPSHUGLFDUGDCHFRGD
QDLQVDDFRVHRRXHLSHDQHGQXHGOOUQQHPLQRHRRHVFFRVGRVODHOJSGPDORRQXHHVHRVLDR
OHDURHODOHRSOHLFDQQULD

Contabilizamos ahora en una tabla las que aparecen con frecuencia mayor, estas deben coincidir con las cifras de la A, B y O.

La posición relativa que ocupa la A en cada tabla será la letra de la clave que buscamos





RESULTADO:

WORLD

c) Muestren el texto descifrado.

FERNANDOALONSODIAZINICIOSUCARRERAENELMUNDODELMOTORDESDEMUYJOVENVIENDOASUID
OLODELAINFANCIAAYRTONSENNAFUEENTONCESCUANDOSUPADRELEREGALOUNKARTQUEELMISMO
HABIACONSTRUIDOPARASUHIJAENUNPRINCIPIOAUNQUEAELLANOLEINTERESOELVEHICULOPORQUE
ERAROTAXYPORESOACABOENMANOSDEFERNANDOCONTANSOLOTRESANOSENESEANOGANOSUPRI
MERACARRERAYCOMENZOAENTRENARSEDESPUESDELASCLASESDELCOLEGIOJUNTOASUPADREELCU
ALEJERCIODEMANAGERYMECANICOUNANOMASTARDEOBTUVOLALICENCIAOFICIALDELAREALFEDER
ACIONESPANOLADEAUTOMOVILISMOENCONSIETEANOSOBTUVOSUPRIMERTITULOENUNACOMPETI
CIONOFICIALELCAMPEONATOINFANTILDEASTURIASZZZGANANDOLASOCHOCARRERASDELASQUECO
NSTABALAPRUEBAUNANOMASTARDEENVOLVIOAPROCLAMARSECAMPEONDEKARTSDEASTURIASYG
ALICIADEBIDOALOSCAMBIOSDECATEGORIALAFAMILIANOSEPODIAHACERCARGODELOSGASTOSQUEA
CARREABANLASCARRERASCERCADELABANDONOELIMPORTADORDEKARTSGENISMARCOSEENCARGO
DELAFINANCIACIONPROPORCIONABALOSKARTSBUSCABAPATROCINADORESOEJERCIAELMISMOCOM
OTALSALVANDOLACARRERADEPORTIVADEFERNANDOALONSOENSEPROCLAMACAMPEONDEASTURI
ASYDELPAISVASCOENCATEGORIACADETESALCANZANDOELSUBCAMPEONATODEESPANAAUNSINPAR
TICIPARENTODASLASCARRERASFUECAMPEONDEESPANAENCOMOJUNIORLOQUELEPERMITIOCOMPE
TIRENELCAMPEONATODELMUNDOALANOSIGUIENTEBECADOPORLAREALFEDERACIONESPANOLADEA
UTOMOVILISMOENDICHOCAMPEONATOQUEDOTERCEROENSEPROCLAMOCAMPEONDEESPANADELT
ROFEOESTIVALDEITALIADELMARLBOROGRANDPRIXYDELCAMPEONATODELMUNDOJUNIORALANOSI
GUIENTEVENCEENLOSCAMPEONATOSDEESPANAITALIAYEUROPAENLACATEGORIAINTERNACIONALA
ENVUELVEAGANARELCAMPEONATODEESPANAELTROFEOPARISBERCYELCAMPEONATOINDUSTRIADE
ITALIA

d) Indiquen la clave de cifrado.

La clave de cifrado es WORLD.

9) Implementen en Python una herramienta que automatice lo máximo posible el criptoanálisis usando Kasiski.

a) Expliquen el mecanismo de funcionamiento del algoritmo.

Inicialmente el algoritmo busca un conjunto de 3 o más caracteres repetidos a lo largo del texto. Cuenta el número de veces que aparecen dichos fragmentos y él y calcula el máximo común divisor (mcd) de las distancias entre los diferentes fragmentos repetidos. Si el mcd es 1, este valor es descartado, puesto que este valor es el que se va a tomar como longitud de clave. Posteriormente se escoge el mcd que tenga mayor número de apariciones para los diferentes fragmentos y se elige como la longitud de clave.

Una vez conocida la longitud de clave lo único que queda por hacer es dividir el texto carácter a carácter, utilizando la longitud de clave para unirlos en los textos cifrados en cada uno de los alfabetos y aplicar análisis de frecuencia en cada uno de los textos para averiguar el carácter que corresponde a la clave. Una vez conocida la clave se realiza el descifrado.

b) Guarden el código implementado como `kasiski.py` (o `kasiski.zip`, en caso de que se utilice más de un archivo).

- El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `kasiski.py`

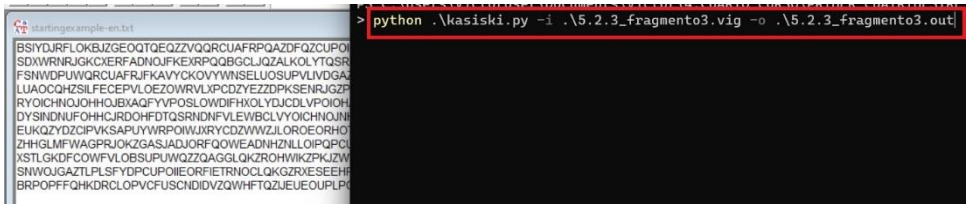
El código se encuentra en el fichero “kasiski.py”.

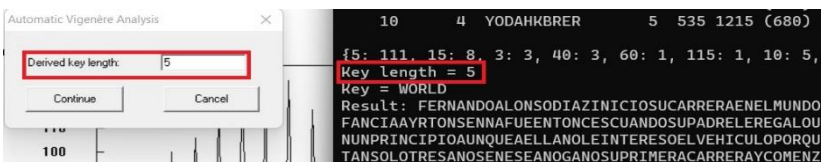
c) Indiquen la sintaxis de uso del comando `kasiski.py`.

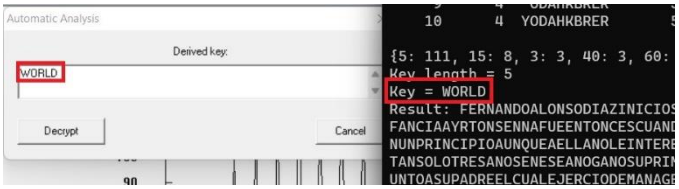
python .\kasiski.py -i .\5.2.3_fragmento3.vig -o .\5.2.3_fragmento3.out

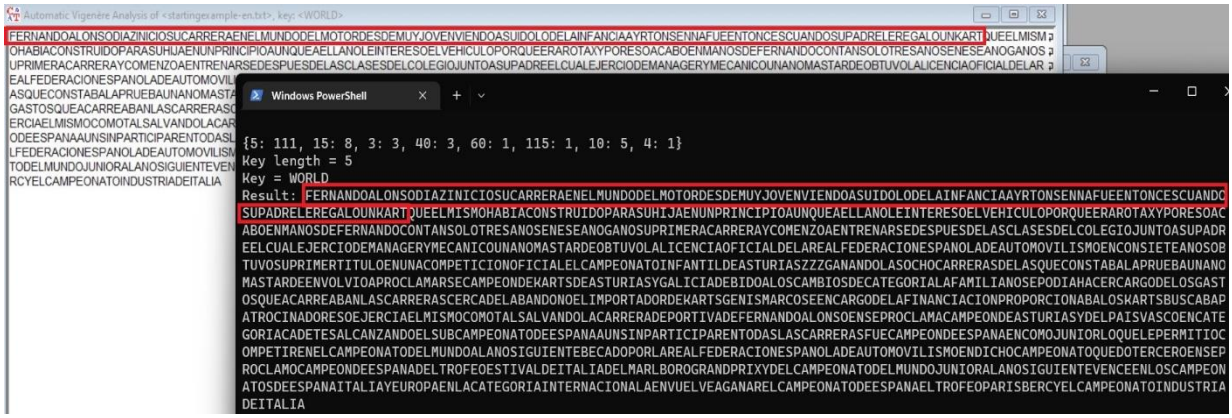
d) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

Al analizar con *Cryptool* el texto “5.2.3_fragmento3.vig”, nos da el mismo resultado que en nuestra implementación






Automatic Analysis
Derived key:
WORLD
Decrypt
Cancel
10 4 YODAHKBRER
{5: 111, 15: 8, 3: 3, 40: 3, 60:
Key length = 5
Key = WORLD
Result: FERNANDOALONSODIAZINICIOS
FANCIAAYRTONSENNAFUEENTONCESCUAND
NUNPRINCIPIOAUNQUEAELLANOLEINTERE
TANSOLOTRESANOSENESEANOGANOSUPRIN
UNTOASUPADREELCUALEJERCIODEMANAGE


Windows PowerShell
{5: 111, 15: 8, 3: 3, 40: 3, 60: 1, 115: 1, 10: 5, 4: 1}
Key length = 5
Key = WORLD
Result: FERNANDOALONSODIAZINICIOSUCARRERAENELMUNDODELMOTORDESDEMUYJOVENVIENDOASUIDOLODELAINFANCIAAYRTONSENNAFUEENTONCESCUANDO
SUPADREELEREGALOUNKARTQUEELMISMOHABIACONSTRUIDOPARASUHIJAENUNPRINCIPIOAUNQUEAELLANOLEINTERESOELVEHICULOPORQUEERAROTAXYPORESOAC
ABOENMANOSDEFERNANDOCONTANSOLOTRESANOSENESEANOGANOSUPRIMERACARRERAYCOMENZOAENTRENARSEDESPUESDELASCLASESDELCOLEGIOJUNTOASUPADR
EELCUALEJERCIODEMANAGERYMECANICOUNANOMASTARDEOBTUVOLALICENCIAOFICIALDELAREALFEDERACIONESPANOLADEAUTOMOVILISMOENCONSIETEANOSOB
TUVOSUPRIMERTITULOENUNACOMPETICIONOFICIALELCAMPEONATOINFANTILDEASTURIASZZZGANANDOLASOCHOCARRERASDELASQUECONSTABALAPRUEBAUNANO
MASTARDEENVOLVIOAPROCLAMARSECAMPEONDEKARTSDEASTURIASYGALICIADEBIDOALOSCAMBIOSDECATEGORIALAFAMILIANOSEPODIAHACERCARGODELOSGAST
OSQUEACARREABANLASCARRERASCERCADELABANDONOELIMPORTADORDEKARTSGENISMARCOSEENCARGODELAFINANCIACIONPROPORCIONABALOSKARTSBUSCABAP
ATROCINADORESOEJERCIAELMISMOCOMOTALSALVANDOLACARRERADEPORTIVADEFERNANDOALONSOENSEPROCLAMACAMPEONDEASTURIASYDELPAISVASCOENCATE
GORIACADETESALCANZANDOELSUBCAMPEONATODEESPANAAUNSINPARTICIPARENTODASLASCARRERASFUECAMPEONDEESPANAENCOMOJUNIORLOQUELEPERMITIOC
OMPETIRENELCAMPEONATODELMUNDOALANOSIGUIENTEBECADOPORLAREALFEDERACIONESPANOLADEAUTOMOVILISMOENDICHOCAMPEONATOQUEDOTERCEROENSEP
ROCLAMOCAMPEONDEESPANADELTROFEOESTIVALDEITALIADELMARLBOROGRANDPRIXYDELCAMPEONATODELMUNDOJUNIORALANOSIGUIENTEVENCEENLOSCAMPEON
ATOSDEESPANAITALIAYEUROPAENLACATEGORIAINTERNACIONALAENVUELVEAGANARELCAMPEONATODEESPANAELTROFEOPARISBERCYELCAMPEONATOINDUSTRIA
DEITALIA

10) Implementen en Python el algoritmo RC4, con las siguientes consideraciones:

- Debe mostrar por pantalla el valor inicial de S, el valor de S después de la fase inicial y cómo va cambiando S con la generación del keystream. La representación de los valores debe hacerse en formato binario.
- La clave debe introducirse en formato hexadecimal
- Para el cifrado:
    - El texto a cifrar se introducirá por consola y se irá cifrando y mostrando el resultado, carácter a carácter, con cada pulsación del teclado.
    - Los caracteres se interpretarán como ASCII.
    - Para cada carácter introducido, se mostrará su codificación en ASCII, en binario, el valor del keystream en binario y el resultado de la operación de cifrado en binario y en hexadecimal.
- Para el descifrado:
    - El texto a descifrar se introducirá por consola en formato hexadecimal.
    - El descifrado se realizará en un solo paso.
    - El resultado se mostrará en formato ASCII.

a) Guarden el código implementado en un archivo denominado `RC4.py`. (o `RC4.zip`, en caso de que utilice más de un archivo).

- El algoritmo debe poder ejecutarse desde una terminal, invocando al comando `RC4.py`

El código se encuentra en el fichero “RC4.py”.

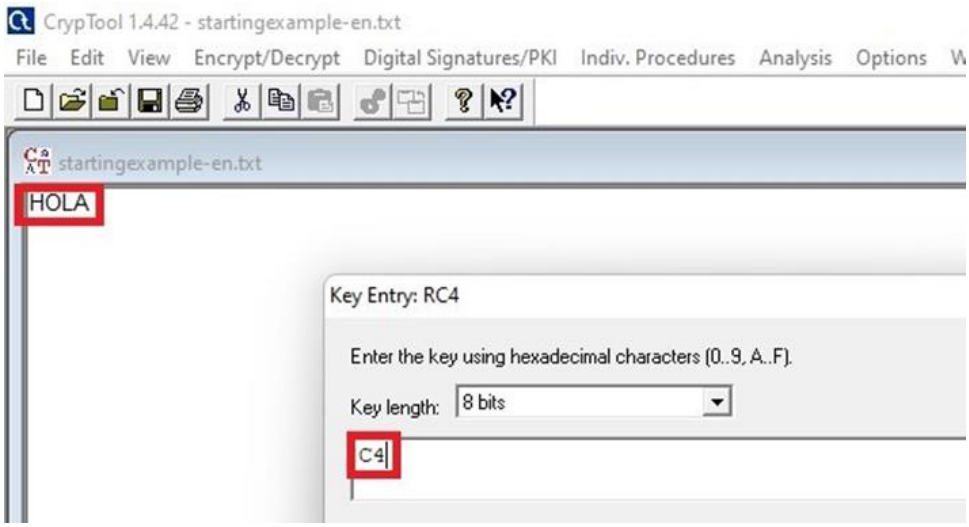
b) Indiquen la sintaxis de uso del comando `RC4.py`.

python RC4.py KEY

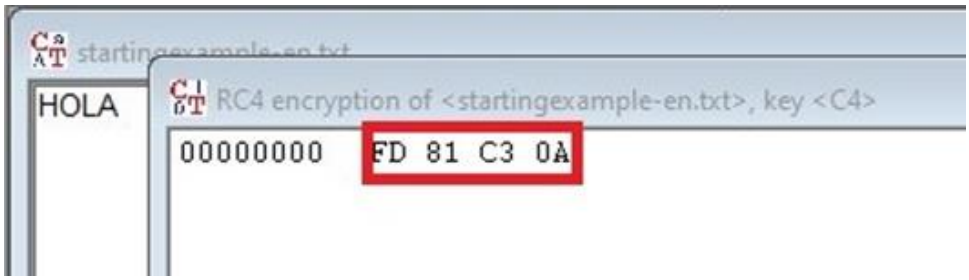
python RC4.py -d KEY

c) Comprueben con la ayuda de Cryptool 1, que su implementación es correcta y muestren evidencia de ello.

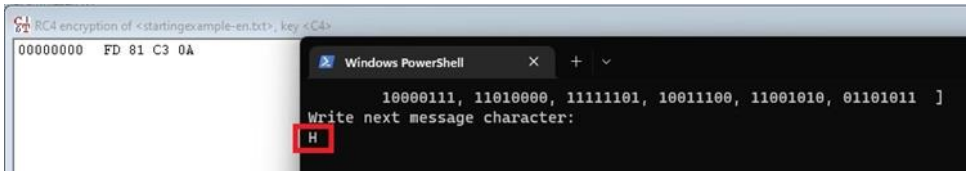
Ejecutamos en *Cryptool* la utilidad de Cifrado de RC4, usando la clave C4.

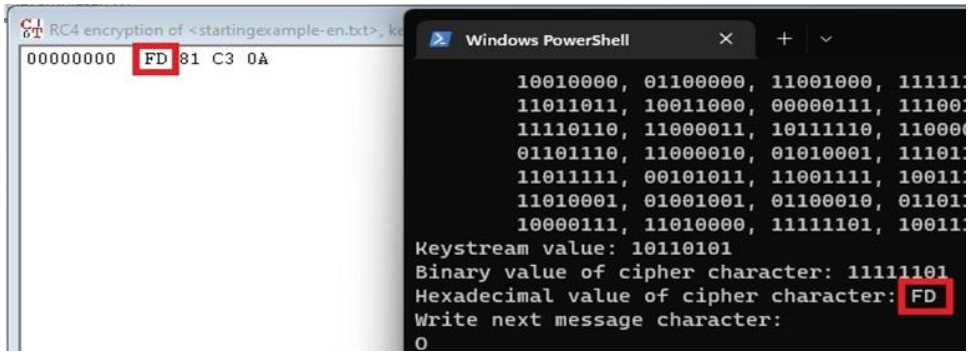


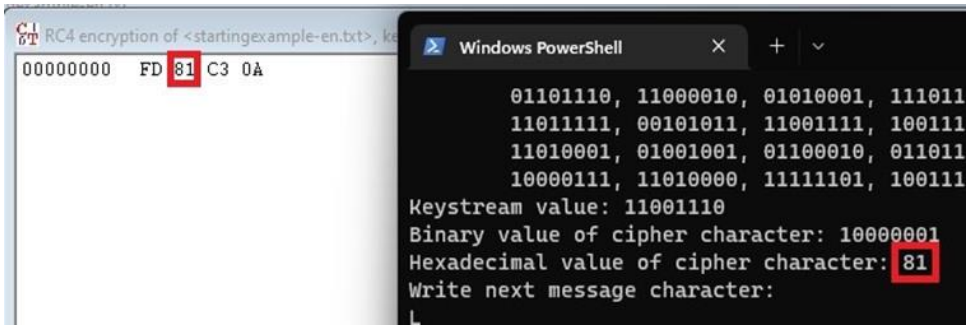
Nos da como resultado el mensaje encriptado FD81C30A.

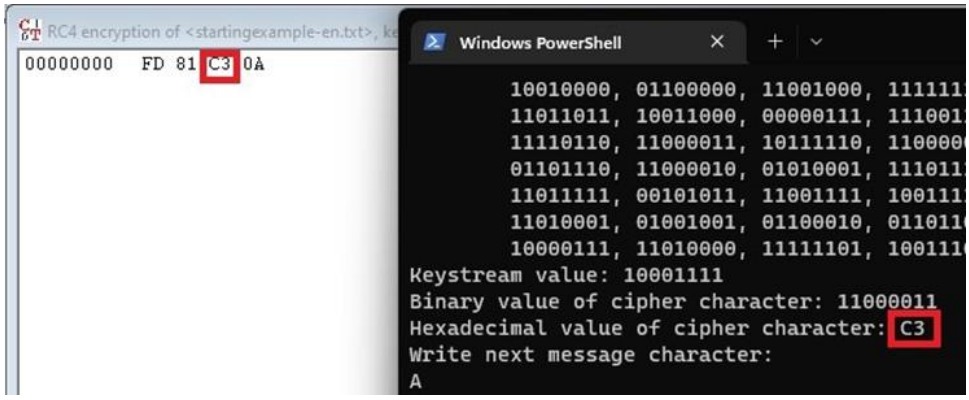


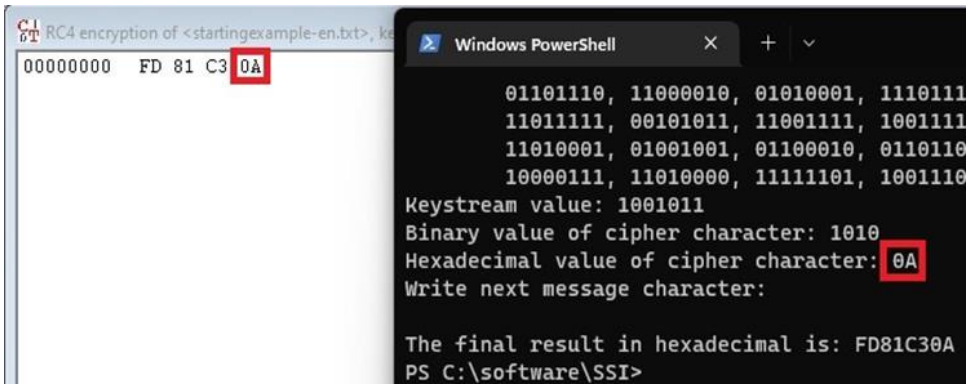
Ahora ejecutamos “RC4.py” con clave C4 (python RC4.py C4) e introducimos uno a uno los caracteres del mensaje “HOLA”. El programa nos mostrará uno a uno los caracteres cifrados en hexadecimal.

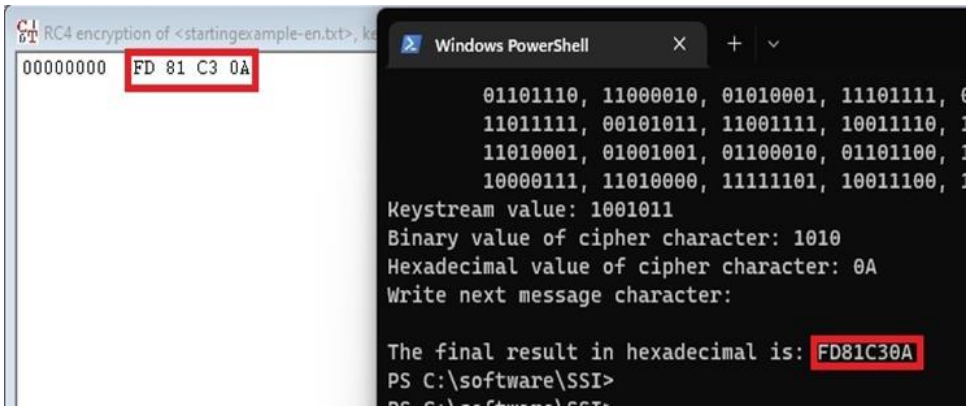




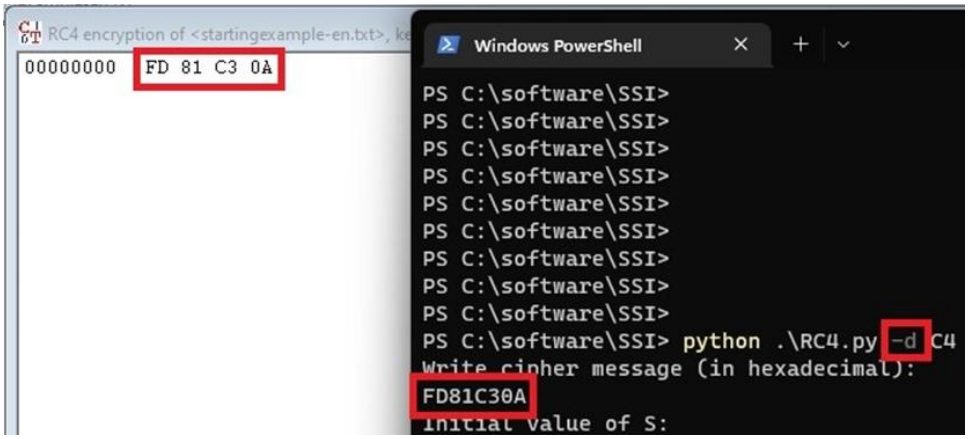








Ahora ejecutamos el script con la opción -d con la misma clave y le pasamos el mensaje cifrado en hexadecimal.



Por último, nos devolverá el mensaje original descifrado.