Criptografía

**Práctica 1. Cifradores clásicos**

“En la presente práctica se llevará acabo la implementación de dos cifradores clásicos, el Cifrado de [Vigenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re) (polialfabético) y el Cifrado Afín (monoalfabético), haciendo uso del Algoritmo de Euclides así como del Algoritmo Extendido de Euclides para este último”

Profesora:

Nidia Asunción Cortez Duarte

Alumno:

García Martínez Kevin

**Algoritmo de Euclides (AE).**

Este algoritmo es una técnica que se emplea para calcular el máximo común divisor entre dos números, en otras palabras, calcular el número entero más grande que divide a ambos números sin dejar residuo. Para el **cifrado afín** se proporciona un **a** y el tamaño del alfabeto **n**, y estos deben de ser **coprimos** entre sí, es decir, el máximo común divisor entre **a** y **n** debe ser **igual a uno**;

**Algoritmo de Extendido de Euclides (AEE).**

Este algoritmo es una extensión del Algoritmo de Euclides que además de calcular el máximo común divisor, nos permite expresarlo como una **combinación lineal**; .  
Este algoritmo será de gran utilidad para el **cifrado afín**, ya que nos permitirá calcular la inversa de **a** que es proporcionada para realizar el cifrado, aunque puede darse el caso que la inversa de **a** (en este caso **x**) sea negativa, por lo que habrá que regresar el valor de la inversa al anillo, esto se logra **sumando múltiplos del módulo** a la inversa hasta que esta sea mayor o igual que cero, ya que sumar múltiplos del módulo es equivalente a **sumar un cero**.

**Código Función AE.**

1. **def** gcd(a,b):
2. **while** b!=0:
3. r = a%b
4. a = b
5. b = r
6. **return** a

**Código Función AEE.**

1. **def** xgcd(a,b):
2. x,y = a,b
3. # gcd(a,0) = a = (a)(1) + 0
4. u0 = 1
5. v0 = 0
6. # gcd(0,b) = b = 0 + (b)(1)
7. u1 = 0
8. v1 = 1
9. **while**(b!=0):
10. q = a//b
11. # r = a - (q)b
12. r = a - (q\*b)
13. # ui = (ui-2) - (q)(ui-1) = u0 - (q)(u1)
14. u = u0 - (q\*u1)
15. # vi = (vi-2) - (q)(vi-1) = v0 - (q)(v1)
16. v = v0 - (q\*v1)
17. # update the values of a, b for the next iteration
18. a = b
19. b = r
20. # update the Bezout coefficients for the next iteracion
21. u0 = u1
22. v0 = v1
23. u1 = u
24. v1 = v
25. **return** {x:u0,y:v0}

**Pruebas con los algoritmos AE y AEE.**

Con

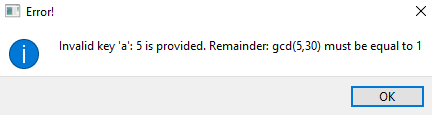


Imagen 1. Prueba del algoritmo AE con a = 5 y n = 30

Con

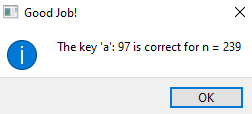


Imagen 2. Prueba del algoritmo AE con a = 97 y n = 239

Con

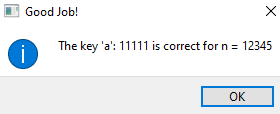


Imagen 3. Prueba del algoritmo AE con a = 11111 y n = 12345

Con

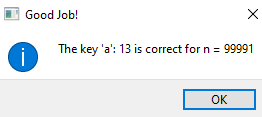


Imagen 4. Prueba del algoritmo AE con a = 13 y n = 99991

Con

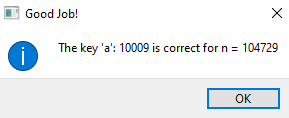


Imagen 5. Prueba del algoritmo AE con a = 10009 y n = 104729

**Cifrador Afín**

**Algoritmo Afín.**

El cifrado afín es un tipo de cifrado por **sustitución simple monoalfabético**, en el que cada símbolo del texto en claro es sustituido por uno del texto cifrado de acuerdo con la siguiente ecuación:

: Es el símbolo **i** del **texto cifrado**.   
: Es la constante de multiplicación.  
: Es el símbolo **i** del **texto en claro**.   
 Es la constante de desplazamiento.   
 Es el tamaño del alfabeto.

Para obtener el texto en claro original es necesario contar con la constante de multiplicación y de desplazamiento **a** y **b** respectivamente, así como el tamaño del alfabeto **n**, con el propósito de aplicar el proceso de **descifrado**, el cual se determina con la siguiente ecuación.:

: Es el símbolo **i** del **texto cifrado**.   
: Es el **inverso multiplicativo** de la constante de multiplicación.  
: Es el símbolo **i** del **texto en claro**.   
 Es el **inverso aditivo** de la constante de desplazamiento.   
 Es el tamaño del alfabeto

**Función de Cifrado Afín.**

1. **def** encryptAffine(plainText, key):
2. cipherText = []
3. a,b,n = key.values()
4. **for** p **in** plainText:
5. c = ( (a \* ord(p)) + b ) % n
6. cipherText.append( chr(c) )
7. **return** "".join(cipherText)

**Función de Descifrado Afín.**

1. **def** decryptAffine(cipherText, key):
2. plainText = []
3. n = key['n']
4. # Reducing the expression for decryp function
5. b = reduceAdditiveInverse(key['a-1'], key['b-1'], n)
6. **for** c **in** cipherText:
7. p = ( (key['a-1']\*ord(c)) + b) % n
8. plainText.append( chr(p) )
9. **return** "".join(plainText)

**Pruebas Cifrador Afín**

**1.** Cifrando el archivo “simon\_paraAffine.txt” con un alfabeto de tamaño de 256 y con un Alpha y Beta proporcionados por el programa.

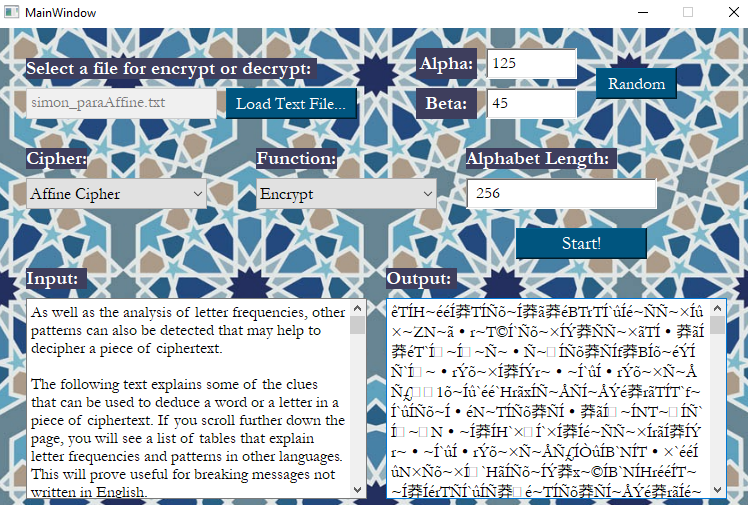


Imagen 6. Muestra de la función de **cifrado** del cifrador afín con el archivo “simon\_paraAffine.txt”

Ahora seleccionaremos el archivo “simon\_paraAffine**.aff**” que genero el programa para poder descifrarlo. Cabe destacar que el programa dependiendo de la extensión que tenga el archivo (.vig o .aff) seleccionará el cifrador que se utilizó para cifrar el archivo original.

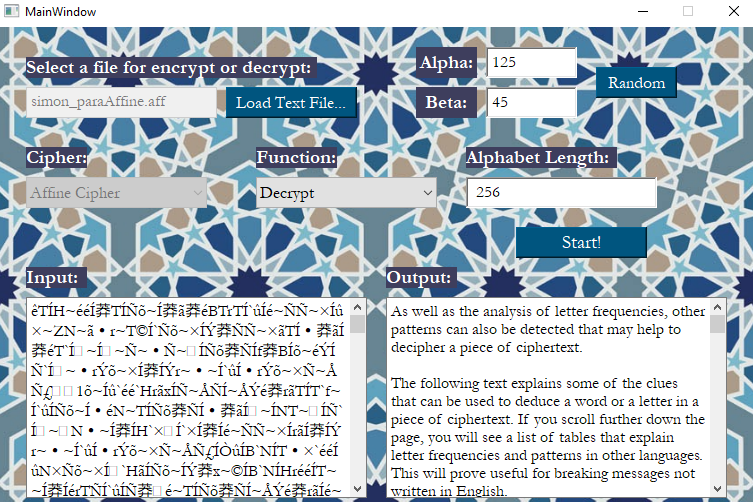
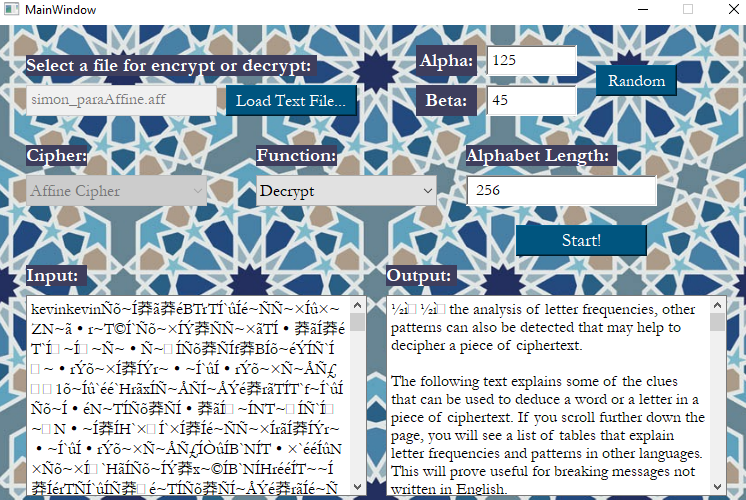
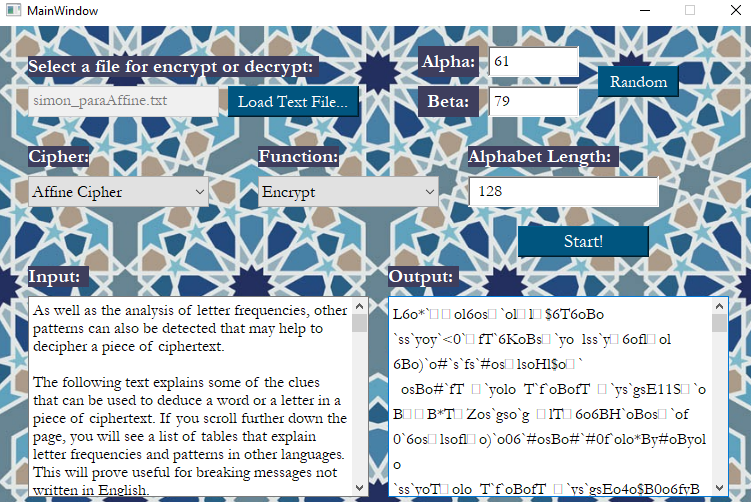


Imagen 7. Muestra de la función de **descifrado** del cifrador afín con el archivo “**simon\_paraAffine.aff**”

Modificaremos los primeros 10 caracteres del texto cifrado por “**kevinkevin**” y volveremos a descifrar.

  
Imagen 8. Modificación del archivo “**simon\_paraAffine.aff**” para descifrar.

**2.** Cifrando el archivo “simon\_paraAffine.txt” con un alfabeto de tamaño de 128 y con un Alpha y Beta proporcionados por el programa.

  
Imagen 9. Muestra de la función de **cifrado** del cifrador afín con el archivo “simon\_paraAffine.txt”

Ahora seleccionaremos el archivo “simon\_paraAffine**.aff**” que genero el programa para poder descifrarlo.

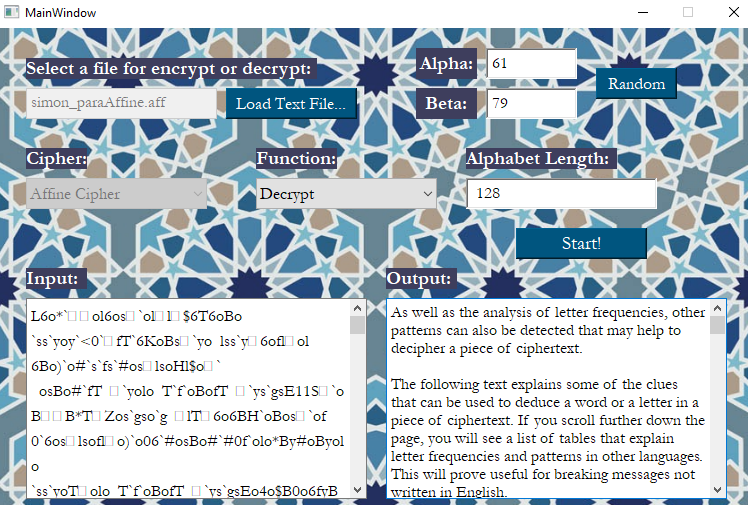
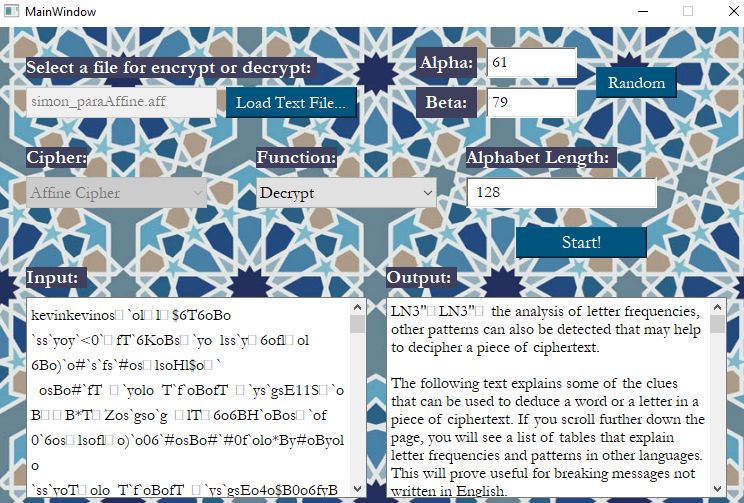
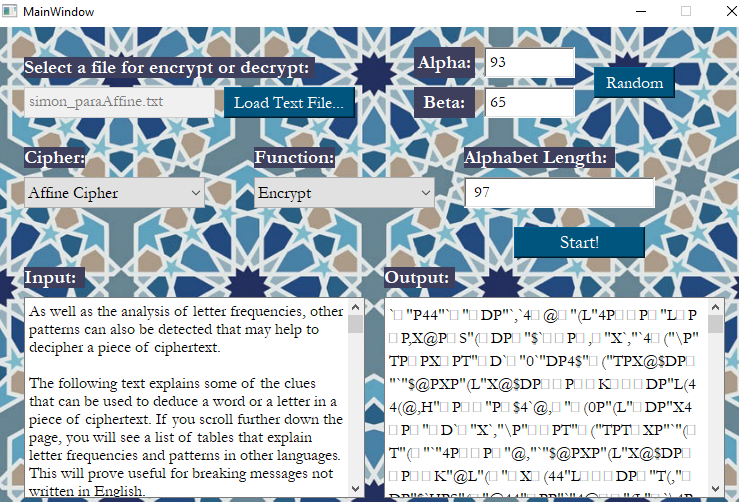


Imagen 10. Muestra de la función de **descifrado** del cifrador afín con el archivo “**simon\_paraAffine.aff**”

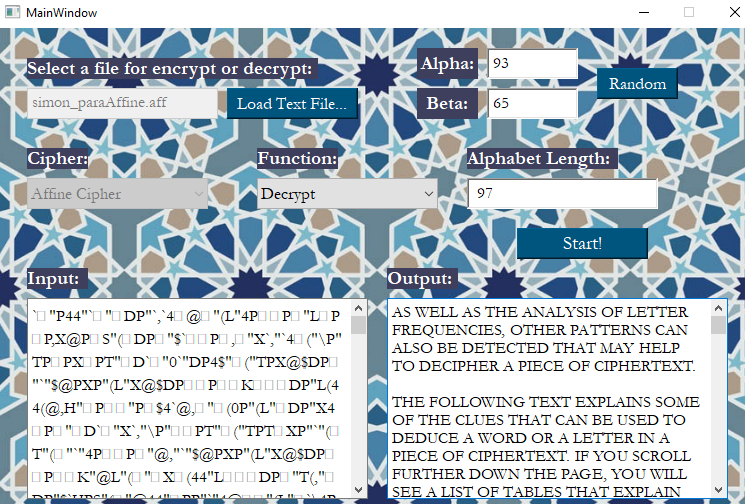
Modificaremos los primeros 10 caracteres del texto cifrado por “**kevinkevin**” y volveremos a descifrar.

Imagen 11. Modificación del archivo “**simon\_paraAffine.aff**” para descifrar

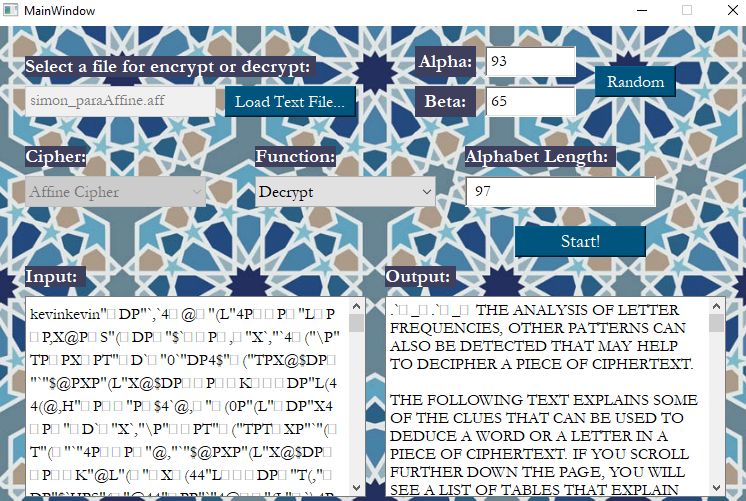
**3**. Cifrando el archivo “simon\_paraAffine.txt” con un alfabeto de tamaño de 97 y con un Alpha y Beta proporcionados por el programa. Es necesario aclarar que para este tamaño de alfabeto todas las letras de la a-z (en minúsculas) se pasaron a mayúsculas, ya que el programa internamente trabaja con el código ASCII de cada letra, y particularmente para el anillo 97, las letras en minúsculas aún no están definidas dentro de este rango.

Imagen 12. Muestra de la función de **cifrado** del cifrador afín con el archivo “simon\_paraAffine.txt”

Ahora seleccionaremos el archivo “simon\_paraAffine**.aff**” que genero el programa para poder descifrarlo, es necesario aclarar que el texto en claro original será mostrado en mayúsculas.

Imagen 13. Muestra de la función de **descifrado** del cifrador afín con el archivo “**simon\_paraAffine.aff**”

Modificaremos los primeros 10 caracteres del texto cifrado por “**kevinkevin**” y volveremos a descifrar.

Imagen 14. Modificación del archivo “**simon\_paraAffine.aff**” para descifrar

**Cifrador V**[**igenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)

**Algoritmo** [**Vigenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)**.**

El cifrado [Vigenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)es un tipo de cifrado por **sustitución simple polialfabético**, es decir, cada símbolo del texto cifrado es obtenido a partir del **alfabeto del texto en claro** y del **alfabeto de la llave**, en donde esta última debe ser del mismo tamaño el texto en claro, en caso contrario, esta deberá repetirse tantas veces como sea necesario para cubrir en su totalidad el texto en claro original. La ecuación para obtener el texto cifrado es la siguiente:

: Es el símbolo **i** del **texto cifrado**.   
 : Es el símbolo **i** de la **llave** utilizada.   
 : Es el símbolo **i** del **texto en claro**.   
 Es el tamaño del alfabeto.

Para obtener el texto en claro original basta con obtener el **inverso aditivo** de cada uno de los símbolos que forman a la llave, de tal modo que , una vez hecho esto, se aplica la siguiente ecuación para realizar el proceso de **descifrado**:

: Es el **inverso aditivo** del símbolo **i** de la **llave** utilizada.

**Función de Cifrado** [**Vigenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)**.**

1. **def** encryptVigenere(plainText, key):
2. cipherText = []
3. n = key['n']
4. x = key['x']
5. **for** (p,k) **in** list(zip(plainText, key['key'])):
6. c = ( ord(p) + ord(k) ) % n
7. cipherText.append( chr(c+x) )
8. **return** "".join(cipherText)

**Función de Descifrado** [**Vigenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)**.**

1. **def** decryptVigenere(cipherText, \_key):
2. plainText = []
3. n = \_key['n']
4. # Shifting + 97
5. x = \_key['x']
6. # Computing the key for decrypt function
7. key = additiveInverseVig(\_key['key'], n)
9. **for** (c,k) **in** list(zip(cipherText, key)):
10. p = ( ord(c) + ord(k) ) % n
11. plainText.append(chr(p+x))
13. **return** "".join(plainText)

**Función para el cálculo de la llave de descifrado para** [**Vigenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)**.**

1. **def** additiveInverseVig(key, n):
2. # Additive inverse of every single character from the provided key
3. addKey = []
4. **for** k **in** key:
5. addKey.append( chr( n-ord(k) **if** n>ord(k) **else** validateInverse(-ord(k),n) ) )
6. **return** "".join(addKey)

**Pruebas Cifrador V**[**igenère**](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)

**1.** Cifrando el archivo “Elvis\_paravigenere.txt” con un alfabeto de tamaño de 26 y con una clave de al menos 8 caracteres. Cabe aclarar que la clave puede ser ingresada por el usuario o ser generada por el programa.

  
Imagen 15. Muestra de la función de **cifrado** del cifrador V[igenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)con la clave “**kevingarciamartinez**”

Ahora seleccionaremos el archivo “Elvis\_paravigenere**.vig**” que genero el programa para poder descifrarlo, al estar manejando el anillo 26 los **espacios en blanco** y los **saltos de línea** (\n) son ignorados, por lo que al descifrar el archivo el contenido de este estará junto y sin saltos de línea, tal y como se muestra a continuación:

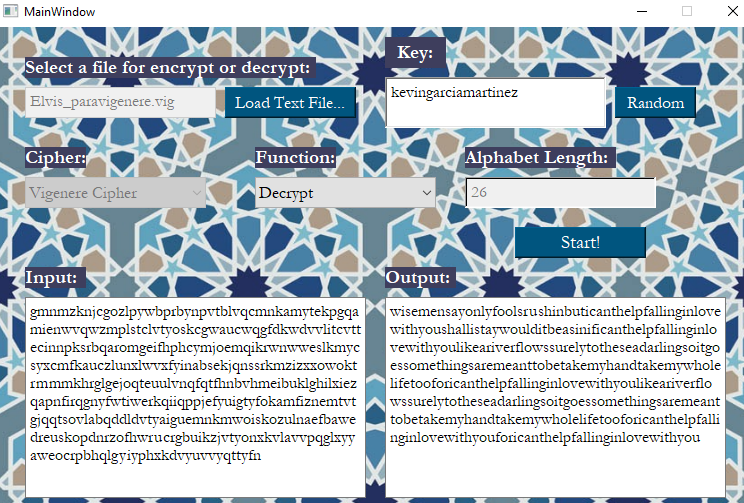
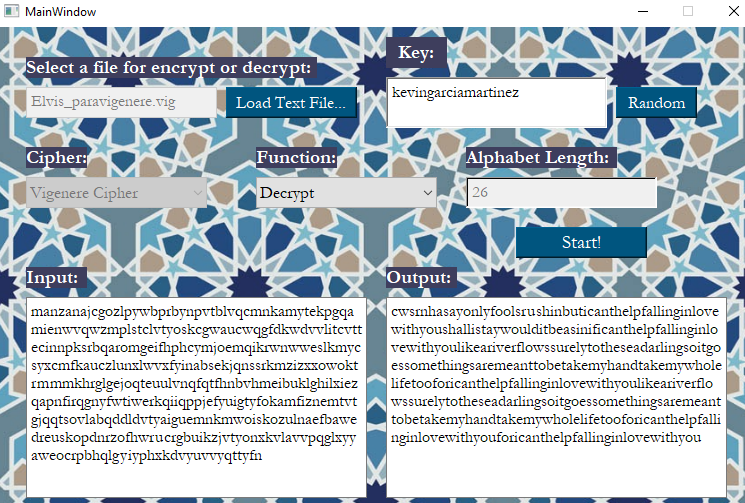
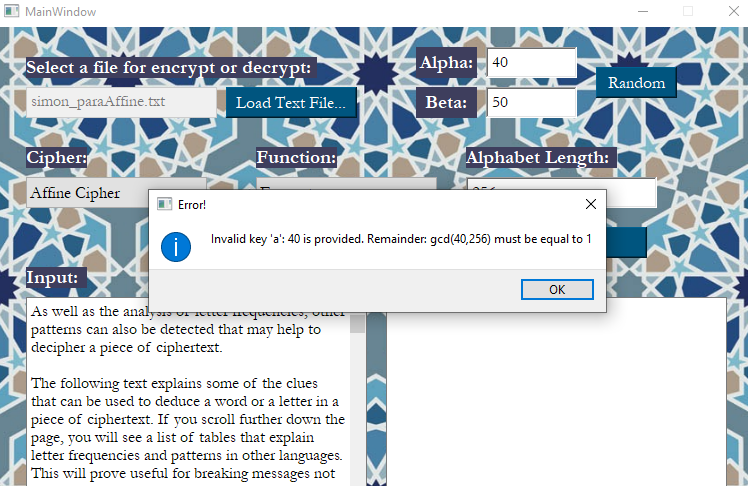


Imagen 16. Muestra de la función de **descifrado** del cifrador V[igenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re)con la clave “**kevingarciamartinez**”

Ahora modificaremos los primeros 7 caracteres por “**manzana**” del texto cifrado y volveremos a descifrar.

  
Imagen 17. Modificación del archivo “**Elvis\_paravigenere.vig**” para descifrar

En el caso del cifrador afín, si se proporciona un valor para **Alpha** que no es coprimo al tamaño del alfabeto **n**, el programa mostrará el siguiente aviso:

Imagen 18. Mensaje de error en caso de que **Alpha** no sea coprimo a **n**.

**Conclusión**

A la hora de trabajar con distintos tamaños de alfabeto n se presentó un problema el cual era que las letras de la **a-z** no estaban definidas para ese anillo en particular, esto se debió a que internamente el programa trabaja con el código ASCII de cada letra que se lea del archivo que se quiere cifrar, por lo que al aplicar con un **n** que sea menor o igual a **65** a la hora de descifrar el archivo no veremos letras, ya que las letras están definidas en el código ASCII a partir del **65** en adelante, por lo que para visualizarlas se tendrían que implementar incrementos y decrementos para no modificar el contenido del archivo.

En el caso del cifrador V[igenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re) se trabajó exclusivamente con 26 (tamaño del alfabeto inglés), se tuvieron que ignorar los espacios en blanco y los saltos de línea, así como también se tuvo que restar **97** a el código ASCII de cada letra del texto original así como a cada letra de la llave para cifrar, con el objetivo de regresar los valores al **anillo 26**, una vez hecho esto a la hora de descifrar se sumó **97** nuevamente al código ASCII de cada letra que se obtuvo de aplicar la ecuación de descifrado, logrando así la correcta visualización del contenido del archivo original.

En el caso del cifrador Afín, para trabajar correctamente con un tamaño de alfabeto igual a 97, todo el contenido del archivo de texto original se paso a mayúsculas, ya que en el código ASCII las letras minúsculas están definidas a partir de 97 en adelante, esto se hizo para obtener el contenido del archivo a la hora de descifrar sin ninguna modificación mayor (caracteres que no sean letras o signos de puntuación).

Tanto el cifrado Afín y el cifrado V[igenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re) son cifradores por sustitución simple, pero el **cifrado Afín es monoalfabético**, lo que implica que cada letra del texto en claro será sustituida por una determinada letra del alfabeto de cifrado, es decir, siempre se utiliza el mismo alfabeto, a diferencia del cifrado V[igenère](https://es.wikipedia.org/wiki/Cifrado_de_Vigen%C3%A8re), el cual es polialfabético, en el cual se utilizan dos alfabetos distintos, el primero corresponde al alfabeto de cifrado y el segundo corresponde al de la llave, pero ambos cifradores siguen el mismo principio, el cuál es sustituir cada letra/carácter del texto en claro por una letra/carácter del alfabeto de cifrado realizando una operación de sustitución.



**Práctica 1. Cifradores clásicos**

|  |  |  |
| --- | --- | --- |
| Documento |  | tú |
| La portada tiene: nombre completo, materia, nombre de profesor, fecha, logotipos, título de práctica y un resumen. (ver ejemplo anexo) | 1 | **1** |
| En media cuartilla con tus palabras explicar el AE y AEE, qué es y para qué sirve cada uno de ellos. | 1 | **1** |
| Código correspondiente a las dos funciones AE y AEE y captura de pantalla de las ejecuciones de las pruebas solicitadas. | 2 | **2** |
| Media cuartilla con tus palabras sobre el algoritmo Affine | 1 | **1** |
| Código correspondiente a las dos funciones (cifrado y descifrado) así como el cálculo de la llave de descifrado y captura de pantalla de las ejecuciones de las pruebas solicitadas. | 2 | **2** |
| Media cuartilla con tus palabras sobre el algoritmo Vigenère | 1 | **1** |
| Código correspondiente a las dos funciones (cifrado y descifrado) así como el cálculo de la llave de descifrado y captura de pantalla de las ejecuciones de las pruebas solicitadas. | 2 | **2** |
| Conclusiones en donde se exprese principalmente las dificultades de la implementación de la práctica (si es que las hubo) así como una reflexión sobre la diferencia entre cifradores de sustitución mono alfabética y poli alfabético. | 1 | **1** |
| El código tiene formato (sugiero utilizar http://www.planetb.ca/syntax-highlight-word) | 1 | **1** |
| Todas las imágenes en el documento tienen título y se referencian en alguna parte del mismo.  (Ej. “en la imagen 1 se muestra …”) | 1 | **1** |
| Programa |  |  |
| El programa cuenta con interfaz gráfica que le permita al usuario elegir la opción deseada: algoritmo y cifrado o descifrado y que reciba los parámetros (en caso de ser necesario) | 2 | **2** |
| La interfaz permite seleccionar el archivo que se va a cifrar/descifrar | 2 | **2** |
| Su programa genera el archivo cifrado con el mismo nombre del archivo de entrada más la extensión .vig o .aff | 2 | **2** |
| Las funciones AE y AEE mandan mensaje “prueba con otro valor” en caso de que alpha no sea coprimo con n | 1 | **1** |
| TOTAL | 20 | **20** |

García Martínez Kevin