**1.Sustitución Monoalfabeto**

**i)Método Afín**

**INTRODUCCION**

Para la realización de este apartado se nos pedía elaborar una rutina en C llamada *afín* con la cual pudiéramos cifrar y descifrar mensajes mediante cifrado afín dada una clave (A,B). Además se debía utilizar la librería de C *GMP*, la cual sirve para realizar operaciones aritméticas con grandes números de forma eficiente.

El programa será invocado de la siguiente manera :

*afin [-C(cifrar)|-D(descifrar] [-m M] [-a A] [-b B] [-i in] [-o out]*

donde (A,B) es la clave, M es el tamaño del espacio del texto cifrado, *in* es el fichero de entrada y *out* el de salida.

**DESARROLLO**

La primera comprobación que se hace en el código es ver que los valores A,M sean ambos menores que M, como se puede ver a continuación:



**ALGORITMOS Y FUNCIONES**

La mayoría de líneas de código de esta rutina se corresponden con la función ***euclidesExtendido***, la cual implementa el algoritmo de Euclides extendido. La función recibe los siguientes parámetros:

-a , b : Entrada de dos números enteros (en formato mpz\_t)

-mcd: Entero en formato mpz\_t donde se escribirá el mcd entre a y b.

-s , t : son dos valores enteros en formato mpz\_t donde se escribirán los dos valores s,t tales que

El pseudocódigo del algoritmo es el siguiente:



En nuestro caso, la entrada a,b se corresponderá con los valores A,M. Lo que necesitamos es que el mcd(A,M)=1, para que a función afín que se define sea inyectiva y se pueda llevar a cabo el descifrado. Además como mdc(A,M)=1, este algoritmo también deja ver quien es el inverso de A en :

y como sabemos que para todo K natural tenemos que

por lo que s es el inverso de A en , valor necesario y por tanto útil para el descifrado.

**2.Sustitución Polialfabeto**

**i)Método de Hill**

**ii)Método de Vigenere**

**3.Método de transposición**

**4.Producto de criptosistemas permutación**

**INTRODUCCION**

Para este apartado se nos pedía realizar un producto de criptosistemas permutación, el cual recibe un texto plano y lo devuelve cifrado. Este cifrado consiste más bien en una desordenación de los caracteres que es única para cada clave, ya que las claves son en sí los vectores permutación que multiplicaremos por la matriz de caracteres para desordenarla. La rutina será invocada de la siguiente manera (ejemplo):

permutacion –C –k1 (2,3,1) –k2 (2,3,4,1) –i input.txt –o output.txt

con lo que tendríamos una clave de dimensión 3 y una clave de dimensión 4, siendo las matrices con las que vamos a trabajar de dimensión 3x4. Es **importante** que las claves se den en el formato especificado y no en otro (paréntesis incluidos).

**DESARROLLO**

Lo primero que hace nuestro programa es leer las claves. Una vez hallado el tamaño de cada una y leídos los valores que contienen, se procede a reservar memoria para las dos matrices P y C (de dimensiónes MxN). Después independientemente de si ciframos o desciframos lo que va haciendo es leer el fichero de entrada carácter a carácter e ir rellenando una matriz MxN con los mismos. Cuando la matriz está llena, es pasada a la función correspondiente (cifrar o descifrar según convenga) y el resultado se manda al fichero de salida en forma de caracteres otra vez. Cuando se acaba el fichero, o bien el programa termina, o si queda alguna matriz a medio llenar la rellena con espacios y entonces termina.

**ALGORITMOS Y FUNCIONES**

Para este programa hemos implementado una librería para manejar matrices M\*N de enteros en C. Los prototipos de las funciones están definidos en *matrixmn.h* y el código de las mismas en *matrixmn.c*. Esta librería nos da soporte para crear (reservar memoria e inicializar a 0 los valores), cambiar valor de una fila y una columna, obtener valor de una fila y una columna, intercambiar dos filas, intercambiar dos columnas, imprimir la matriz por pantalla, etc.

 La estructura de datos *matrixmn* queda definida en la cabecera de la siguiente manera:

Con este tipo de datos podemos realizar las siguientes operaciones básicas:



Además, en esta librería están las funciones que se encargan del cifrado y el descifrado en sí (llamadas *cifra* y *descifra*). La función *cifra* recibe una matriz de con caracteres del texto plano (ordenados) y las dos claves (como arrays de enteros) para devolver una matriz de la misma dimensión y con los mismos elementos, pero ya cifrados (desordenados en este caso). Para ello primero permuta las filas de la matriz origen y luego las columnas, según el orden que le indiquen las claves. La función *descifra* hace lo mismo pero de forma contraria, empezando por las columnas y acabando con las filas para convertir la matriz C cifrada en otra P descifrada.