

Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Asignatura

**INFORMÁTICA**

Curso 2016-2017



Universidad Politécnica de Madrid

Grado en Ingeniería Electrónica y Automática

Datos del Grupo

Eduardo Gibert Almela 54627

David Tertre Boyé 54882

# Introducción

Vamos a hacer un programa que sirva para descifrar una contraseña numérica escrita en un fichero. Además informará del tiempo que ha tardado en hacerlo.

# Desarrollo

Para llevar a cabo las funciones del programa, se define una dimensión N=9 para el número de dígitos de la contraseña, ya que pasados los 10 se alargan demasiado las operaciones.

El programa cuenta con dos funciones aparte de la principal.

**La primera (*cambiocontraseña*)**, permite al usuario entrar en una interfaz en la que se le permite renovar la contraseña y se le muestran los requisitos de ésta.

Se utilizan tres bucles “do-while”, cada uno dentro de otro, para verificar las condiciones de cada dígito introducido.

Para registrar la clave se usan los punteros de la contraseña y de la contraseña validada (la segunda que se introduce para confirmar la primera). Se crean estos punteros para poder hacer la asignación dinámica de memoria y posteriormente limpiarla.

Los dígitos son mostrados en pantalla como asteriscos y en caso de que el usuario presione la tecla de retroceso, se eliminará el último carácter introducido (se sustituye por ‘\0’) y se borrará un asterisco. Para finalizar la escritura de la clave el usuario presiona ENTER, lo que provoca el cierre del primer bucle.

A través de un bucle “for” se verifican las condiciones de la contraseña y en caso de fallar, vuelve a comenzar el bucle principal, en caso de acertar, pasa al siguiente bucle “do-while” que pide la contraseña por segunda vez, para asegurar que el usuario la recuerda. Si coinciden, se cierra al fin el bucle principal y la contraseña es actualizada, en caso contrario, la función se reinicia.

**La segunda función (*descifrador*)** es mucho más simple.

Tan sólo cuenta con un bucle “for”, cuya variable comienza en 0 hasta 10 elevado a la dimensión N. El condicional “if” compara el número de la variable con el de la contraseña. Si son distintos, se le suma uno y continúa, y si son iguales, se cierra el bucle con el valor último de la variable: la contraseña descifrada.

**En la función *main***, a parte de las variables, se declaran los ficheros. Uno de ellos sirve para escribir el código a descifrar y el otro para mostrar la clave ya decodificada con el tiempo empleado en hacerlo.

Se muestra al usuario el menú principal donde se permiten tres opciones: cambiar la contraseña, descifrarla o salir del programa.

En el primer caso, se inicia la función “cambiocontraseña” y una vez cambiada la clave, se abre el fichero que la contiene. En caso de estar dañado el archivo, se muestra un mensaje de “apertura incorrecta” y se regresa al menú principal. Si en cambio se accede con normalidad al fichero, se actualiza la contraseña en éste y se cierra para volver al menú principal.

En el segundo caso, abre el fichero de contraseña y realiza un bucle de lectura del fichero para guardar la clave y compararla posteriormente.

Abre el fichero de resultado sobre el que se va a escribir.

Comienza a cronometrar, captando el instante actual con la función clock() (de la biblioteca time.h) y guardándolo en la variable “comienzo”.

Inmediatamente después se inicia la función de descifrador y cuando ésta finaliza, se graba nuevamente el instante actual en la variable “final”. El tiempo de decodificación es por lo tanto igual a “final” – “comienzo”.

La contraseña decodificada y el tiempo ocurrido quedan grabados en el fichero de resultado y se imprimen por pantalla.

Se cierra el fichero y se regresa al menú principal.

Por último, en el tercer caso, se muestra un mensaje de “Hasta la vista!” y el programa finaliza.

Si no se elige ninguno de estos tres casos, se muestra un mensaje de “Seleccione una opción posible”.

# Conclusiones

A diferencia de lo que creíamos, ha sido mucho más laborioso crear una función que permita registrar una contraseña, con unos valores determinados, que una función que se dedique a decodificarla. Nos ha costado bastante pensar cada bucle y función, pero el resultado ha valido la pena en nuestra opinión.

# Bibliografía

Libro “Introducción a la programación en C”.

Apuntes tomados en clase.

Tutorialspoint.com (para la biblioteca time.h).