Introducción a Python. Ejercicios (III)

1. El Código Cuenta Cliente (CCC) es el código que identifica en España las cuentas corrientes de los clientes de bancos. El CCC tiene 20 dígitos en formato AAAA-BBBB-CC-DDDDDDDDDD.

   - AAAA son cuatro dígitos que identifican la entidad bancaria.
   - BBBB son cuatro dígitos que identifican la oficina.
   - CC se denomina dígito de control (DC).
   - DDDDDDDDDD son 10 dígitos de la cuenta del cliente en el banco.

   Según la Wikipedia:

   > Los dígitos situados en las posiciones novena y décima se generan a partir de los demás dígitos del CCC, permitiendo comprobar la validez del mismo y reducir la posibilidad de errores de manipulación. El primero de ellos valida conjuntamente los códigos de entidad y de oficina; el segundo, valida el número de cuenta.

   Cada uno de los dígitos del DC se calcula utilizando el mismo algoritmo, para lo que se complementa con dos ceros a la izquierda la entidad y oficina.

   La siguiente función en Python calcula el DC correspondiente para una lista de 10 número enteros:

```
def calcula_dc(lista):
    """Calcula el dígito de control de una CCC.
    Recibe una lista con 10 numeros enteros y devuelve el DC
    correspondiente"""

    pesos = [1, 2, 4, 8, 5, 10, 9, 7, 3, 6]
    aux = []
    for i in range(10):
        aux.append(lista[i]*pesos[i])
    resto = 11 - sum(aux)%11
    if resto == 10:
        return 1
    elif resto == 11:
        return 0
    else:
        return resto
```

   Por ejemplo:

```
>>> lista = [1, 6, 7, 0, 0, 0, 0, 3, 3, 2]
>>> calcula_dc(lista)
5
```

   Escribe un programa que pida al usuario un CCC en el formato arriba indicado y compruebe su validez.

2. Añade al programa anterior la verificación de la entidad bancaria, para ello se debe pedir al usuario el nombre de la entidad financiera y contrastarla con las entidades del fichero bancos.txt.

3. Implementa un sistema completo de validación de usuarios en una máquina con Debian squeeze, que tiene las siguientes características:

   - El nombre de usuario y las contraseñas se almacenan en el fichero /etc/shadow, al que tiene acceso sólo el usuario root.
   - Las contraseñas se almacenan como un hash AES512 con sal

   Ayuda:

   Supongamos que tenemos en nuestro sistema el usuario `prueba` con contraseña `asdasd`, una línea correspondiente a este usuario en el fichero /etc/shadow sería:

   ```
   prueba:$6$/nNkCgcv$r.FooJSMDwP2gd4MAsoRTTLoOVpsIF2EyxW59ryWW7bpKUx\
   ulWX9CpEWknaDBzHWYJ2q9gqxEyfQl93u7okPa.:15059:0:99999:7::::
   ```

   - La sal de una contraseña cifrada se indica en linux por los 12 primeros caracteres del hash de la contraseña, en el caso anterior la sal sería `$6$/nNkCgcv$`.
   - La función crypt del módulo crypt permite formar los hashes con sal utilizados por linux, de la siguiente manera:

     ```
     >>> from crypt import crypt
     >>> crypt('asdasd','$6$/nNkCgcv$')
     '$6$/nNkCgcv$r.FooJSMDwP2gd4MAsoRTTLoOVpsIF2EyxW59ryWW7bpKUxul\
     WX9CpEWknaDBzHWYJ2q9gqxEyfQl93u7okPa.'
     ```

     donde `asdasd` es la contraseña en claro.

4. Utilizando el ejercicio anterior, crea una aplicación simple de *craqueo* de contraseñas utilizando el ficheros*most_used_pass.txt*, que contiene 10000 contraseñas habituales.

   Alternativamente puedes plantear una solución de "fuerza bruta"que consiste en ir construyendo palabras cada vez más complejas:

   - Todas las palabras con un caracter
   - Todas las palabras con dos caracteres
   - Todas las palabras con tres caracteres
   - ...

   La relación entre el tiempo empleado en romper la contraseña y su longitud es exponencial, por eso se recomienda hoy en día utilizar una contraseña de 10 caracteres o más.