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).
   - DDDD 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.

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

```
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

# Solicitamos al usuario el CCC y almacenamos los cuatro componentes
# en una lista:
ccc = raw_input("Introduce el CCC: ").split('-')
# Transformamos la entidad y la oficina en una lista de 10 numeros:
entidadyoficina = [0, 0]
for i in range(2):
    for j in range(4):
        entidadyoficina.append(int(ccc[i][j]))
# Transformamos la cuenta en una lista de 10 numeros:
cuenta = []
```

```
for i in range(10):
    cuenta.append(int(ccc[3][i]))
# Calculamos el dc para cada lista y los concatenamos:
dc = str(calcula_dc(entidadyoficina))+str(calcula_dc(cuenta))
# Comparamos el DC calculado y el proporcionado:
if dc == ccc[2]:
    print 'CCC verificado'
else:
    print 'Hay un error en el CCC'
```

2. Añade al programa anterior la verificación de la entidad bancaria, para ello se leerá del 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

```
from getpass import getpass
from crypt import crypt

def passwd_check(clear_passwd,crypt_password):
    sal = crypt_password[0:12]
    if crypt(clear_passwd,sal) == crypt_password:
        return True
    else:
        return False

fichero = open('shadow','r')

vueltas = 1
while True:
    usuario = raw_input("login: ")
    contrasenna = getpass("password: ")

    for linea in fichero:
        campos = linea.split(':')
        if campos[0] == usuario:
            if passwd_check(contrasenna,campos[1]):
                print 'Usuario correcto'
                vueltas = 3
            break

    if vueltas == 3:
        break
    else:
        print 'Usuario o contrasenna incorrectos'
        vueltas+=1
        fichero.seek(0)

fichero.close()
```

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.

```
from getpass import getpass
from crypt import crypt

def passwd_check(clear_passwd,crypt_password):
    sal = crypt_password[0:12]
    if crypt(clear_passwd,sal) == crypt_password:
        return True
    else:
        return False

fichero = open('shadow','r')
fichero2 = open('most_used_pass.txt','r')

for linea in fichero:
    campos = linea.split(':')
    if campos[1][0] == '$':
        for linea2 in fichero2:
            if passwd_check(linea2.rstrip(),campos[1]):
                print 'La pass del usuario %s es %s' % (campos[0],\
                linea2.rstrip())
                break
    fichero2.seek(0)

fichero.close()
fichero2.close()
```