Fonaments de Programació (104337)

Curs 2018-19 - Examen Recuperació (1 de febrer de 2019)

Nom estudiant: **NIU:**

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, els procediments i funcions han d'estar ben programats (utilitzant les instruccions més adients, sense operacions ni variables innecessàries, etc.)

1) Volem fer un programa en Python que ens permeti jugar al joc del penjat contra la màquina. La màquina triarà aleatòriament una paraula, i ens mostrarà per pantalla quants caràcters té. L'usuari pot anar dient lletres, i depenent de si la paraula conté o no aquesta lletra es contarà com a error o no. El joc acaba o quan l'usuari ha endevinat totes les lletres que formen la paraula o quan ha comès massa errors.

a) Programar una funció `load_data(filename)`, que s'ha d'encarregar de obrir i llegir un fitxer que conté milers de paraules, una per línia. Ha de retornar una llista amb cada una de les paraules. (**1 punt**)

b) Programar una `funció filter_and_select(list,n)` que donada una llista `list` de paraules d'entrada ha de filtrar aquelles paraules que tinguin menys de `n` lletres diferents (per ex. les paraules `'papa', 'gos', 'mono'`, no s'acceptaran si `n=3`), i de la resta triar-ne i retornar-ne una aleatòriament. Podem emprar la funció `randint(a,b)` del mòdul `random` que ens retornarà aleatòriament un valor enter dins de l'interval `[a,b].` Es bonificara si el filtratge de les paraules es fa emprant una `list comprehension`. (**1 punts**)

c) Definir una funció `check(sol, chars)` que donada una paraula `sol`, i una llista de caràcters `chars`, retorni un valor booleà i una cadena de caràcters. La cadena de caràcters mostrarà els caràcters de `sol` separats per espais que apareixen dins de `chars`, i reemplaçarà la resta per guions '_'. El valor booleà serà `True` si la llista de chars ja conté tots els caràcters diferents de `sol`, `False` si no. (**2 punts**)

```
print(check('python', []))
(False, '_ _ _ _ _ _')
print(check('python', ['o']))
(False, '_ _ _ _ o _')
print(check('python', ['o','p','z','a','n','x','t']))
(False, 'p _ t _ o n')
print(check('python', ['o','p','z','a','n','x','t','y','h']))
(True, 'p y t h o n')
```

d) Escriure el codi principal del programa que segueixi els següents passos:

1. Demanar a l'usuari que entri per teclat el nom del fitxer a llegir i el nombre mínim de caràcters diferents. A continuació llegir el fitxer amb la funció `load_data`
2. Seleccionar la paraula emprant `filter_and_select`
3. Mentre l'usuari no hagi no hagi entrat 10 caràcters erronis o hagi endevinat ja la paraula sencera:
    a. Demanar nou caràcter.
    b. Si ja l'havia entrat, demanar-ne un de nou, fins que n'entri un que no hagi entrat amb anterioritat.
    c. Afegir-lo a la llista que ens servirà per controlar els caràcters entrats.
    d. Comprovar si apareix a la paraula o no. Si no, aumentar el contador de caràcters erronis.
    e. Cridar la funció `check` per tal de poder mostrar per pantalla el progrès. (**2 punts**)

2) Què mostra per pantalla la següent instrucció? (**0.5 punts**)

```
print([(x+y) for x,y in zip(range(3,27,3),range(5,45,5)) if y%2==0])
```

3) Què mostra per pantalla el següent codi? (**0.5 punts**)

```
>>> import numpy as np
>>> a=np.array([[3,2],[3,5],[1,2],[1,1]])
>>> b=np.array([[2,2]])
>>> print(np.sum(np.abs(a-b),axis=1))
```

4) Imaginem que tenim un diccionari d, on les claus son països i els seus valors associats son les capitals d'aquests països. Construir un diccionari d_inv a partir de d, que tingui les capitals com a claus i els països com a valors. Emprant `d_inv`, mostrar per pantalla les deu primeres capitals i països ordenades per capital. (**1 punt**)

5) Definir una funció `Hamming(b1,b2)` que retorni la distància de Hamming entre dos strings `b1` i `b2` que representin seqüències de bits. La distància de Hamming, retorna el nombre de bits diferents entre les dues seqüències, per exemple (**1 punt**)

```
>>> b1='0100101001'
>>> b2='1101101010'
>>> print(Hamming(b1,b2))
4
```

6) Definir la classe `Film`, que tindrà com a atributs el títol de la pel·lícula, nom de director, any d'estrena i una llista dels actors principals.

Apart del constructor sobrecarregarem l'operador __str__ per tal de poder fer prints dels objectes de tipus film i l'operador __ge__ (greater or equal >) que ens permetrà comparar dues pel·lícules. Direm que una pel·lícula és "més gran" que una altra si s'ha estrenat amb anterioritat.

Per últim, definirem una funció PushActor(a) que ens permetrà afegir actors a una certa pel·lícula i PopActor(a), que haurà d'esborrar l'actor de la pel·lícula. (**1 punt**)