Fonaments de Programació (104337)
Curs 2019-2020 - Examen de Recuperació (31 de gener de 2020)

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.)

Els exercicis del 1 al 6 són funcions o procediments que s'avaluen de forma independent, però **les implementacions que en feu han de ser coherents entre elles i amb l'exercici 7**.

Un club de futbol, disposa d'un fitxer de dades que conté la informació dels partits que ha disputat durant un campionat. A cada línia del fitxer, es guarden les dades corresponents a un partit en el següent format:

```
jornada::data::tipus::equip::gols_marcats::gols_rebuts
```

on "jornada" és el número de partit disputat, "data" és la data del partit en format AAAA-MM-DD, "tipus" indica si l'equip ha jugat com a local ('L') o com a visitant ('V'), "equip" és el nom de l'equip contrincant, i "gols_marcats" i "gols_rebuts" són els gols que l'equip ha marcat o rebut en aquell partit. Les dades estan separades per la cadena "::" i al final de cada línia hi ha un salt de línia ('\n'). Per exemple, el contingut que podria tenir el fitxer seria:

```
1::2019-04-11::L::Real Betis Balompie::2::1
2::2019-04-18::V::Granada Club de Fútbol::4::0
3::2019-04-25::V::Athletic Club::2::5
4::2019-05-02::L::Deportivo Alavés::5::3
```

Exercici 1 (1 punt)
Fer una funció anomenada `llegir_partits` que rebi com a paràmetre el nom d'un fitxer (string) i retorni un diccionari amb les dades de tots els partits continguts al fitxer segons el format explicat anteriorment. En el diccionari amb les dades dels partits, cada element tindrà per clau el número de jornada del partit i com a valor, una llista amb les dades del partit segons l'ordre en què estan al fitxer, és a dir, el primer element de la llista serà la data, el segon el tipus de partit (L/V), el tercer el nom de l'equip contrari, etc.

Notes: Recordeu que al final de cada línia del fitxer, hi ha un salt de línia.
La jornada, els gols marcats i els gols rebuts s'han de tractar i guardar com a enters.

Exercici 2 (1 punt)
Fer un procediment anomenat `desar_partits` que rebi dos paràmetres: **partits** (diccionari amb les dades dels partits d'un equip) i **nom_fitxer** (nom del fitxer on s'han de guardar les dades). Aquest procediment ha de guardar les dades dels partits en el fitxer especificat. Les dades dels partits s'han de guardar en ordre segons el número de jornada i seguint el format de fitxer mostrat anteriorment. Si el fitxer ja existeix i conté dades, es sobreescriurà el seu contingut.

Nota: En el diccionari de partits, és possible que faltin les dades d'algunes jornades. Per exemple, pot haver-hi les dades de les jornades 1, 2, 3 i 5, però no les de la jornada 4.

Exercici 3 (1 punt)
Fer un procediment anomenat `afegir_partit` que rebi tres paràmetres: **partits** (diccionari amb les dades dels partits), **jornada** (enter amb el número de jornada per la qual volem afegir un partit), i **nom_fitxer** (nom del fitxer on es guarden les dades dels partits). Aquest procediment ha de comprovar si al diccionari de partits hi ha dades guardades per la jornada especificada. En cas afirmatiu, es mostrarà el missatge "`La jornada X ja conté dades`", on X és el número de jornada i no es farà res més.

Si no hi ha dades per la jornada indicada, s'han de demanar les dades del partit que es vol afegir (data, si s'ha jugat com a local o visitant, equip contrari, gols marcats i gols rebuts) i, a continuació, afegir les dades del nou partit al diccionari segons el format especificat a l'ex.1. Un cop afegides les dades al diccionari, es guardarà la informació dels partits a fitxer amb el procediment `desar_partits` de l'ex. 2.

Nota: Assumim que sempre s'introduiran dades correctes (no cal controlar que els valors entrats siguin vàlids).

Exercici 4 (1 punt)
Fer una funció anomenada `estadistica_gols_rebuts` que rebi com a paràmetre el diccionari de partits i retorni una llista d'enters amb el número de vegades que l'equip ha rebut 0 gols, 1 gol, 2 gols, etc.

Nota: Podeu suposar que en cap partit s'han rebut 10 o més gols.

Exercici 5 (1 punt)
Fer una funció anomenada `filtra_partits` que rebi tres paràmetres: **partits** (diccionari on guardem els partits), **dada** (un string que indica per quina dada volem filtrar) i **valor** (el valor de la dada que volem filtrar). Aquesta funció ha de retornar una llista d'enters que contingui els números de les jornades on la dada indicada té un valor igual al passat com a paràmetre.

Els valors possibles de **dada** són: "LV" per local/visitant i "EQ" per l'equip contra qui es va jugar. Per qualsevol altre string que es passi com a **dada**, s'ha de mostrar el missatge "`Error: Codi inexistent`" i s'ha de retornar una llista buida.

Per exemple, si volem filtrar per la dada "LV" amb valor 'L', la funció retornaria la llista [1,4] indicant que a les jornades 1 i 4 es va jugar com a local.

Nota: La solució només es considerarà completament correcta si alguns passos es fan amb list comprehensions.

Exercici 6 (1 punt)
Fer un procediment anomenat `cerca_per_data` que rebi dos paràmetres: **partits** (diccionari on guardem els partits) i **data** (un string amb la data del partit a cercar). Aquest procediment ha de mostrar la informació (jornada, L/V, equip contrari, gols marcats i gols rebuts) del partit que es va jugar en la data passada com a paràmetre. Si no hi ha cap partit que té una data igual a la que s'ha passat com a paràmetre, mostrar el missatge "`Error: No hi ha cap partit en aquesta data`".

Notes: En una mateixa data no hi ha mai dos partits.

Exercici 7 (2 punts)
Fer un programa que segueixi els passos que teniu a continuació. **Per a fer els passos indicats, necessiteu les funcions i procediments dels exercicis anteriors.**

1. Inicialitzacions de variables i constants.
2. Utilitzar la funció `llegir_partits` de l'ex.1 per a fer la lectura del fitxer `campionat.txt` amb la informació dels partits d'un campionat. Si en la lectura del fitxer es produeix alguna excepció (`FileNotFoundError` o `ValueError`), escriure un missatge d'error (`"Fitxer no trobat"` o `"El fitxer conté dades incorrectes"` respectivament) i acabar el programa. En cas contrari, seguir amb el punt 3.
3. Utilitzar (NO IMPLEMENTAR) el procediment `menu_principal()` que imprimeix per pantalla el següent menú:

```
---- MENU ----
1.- Afegir partit
2.- Estadística de gols
3.- Consultar jornades
4.- Cercar partit per data
5.- Finalitzar
```

4. Demanar a l'usuari que introdueixi una opció del menú.
5. Si l'opció és 1:
   5.1. Demanar a l'usuari que introdueixi per quina jornada vol afegir un partit. Si l'usuari introdueix un valor negatiu o zero, es mostrarà un missatge d'error i es demanarà a l'usuari que torni a introduir el valor de la jornada. Aquest pas es repetirà fins que s'introdueixi un valor positiu.
   5.2. Utilitzar el procediment `afegir_partit` de l'ex.3 per afegir al diccionari de partits i al fitxer `campionat.txt`, les dades d'un partit a la jornada introduïda per l'usuari.
6. Si l'opció és 2, utilitzar la funció `estadistica_gols_rebuts` de l'ex. 4 per a calcular en quants partits l'equip ha rebut 0 gols, 1 gol, 2 gols, etc. i mostrar el resultat per pantalla amb format:

```
Estadística de gols rebuts:
0 – 3 partits
1 – 5 partits
...
9 – 0 partits
```

7. Si l'opció és 3:
   7.1. Demanar a l'usuari sobre quina dada vol fer la consulta ("LV" per local/visitant o "EQ" per l'equip contra qui es va jugar) i el valor que vol consultar.
   7.2. Utilitzar la funció `filtra_partits` de l'ex. 5 per obtenir la llista de jornades on la dada consultada coincideix amb el valor indicat.
   7.3. Mostrar el resultat obtingut amb el missatge: "`Resultat de la consulta: <X1>, <X2>, ...`" on X1, X2,... són els números de les jornades retornades per `filtra_partits`. Si la consulta no ha produït cap resultat, el missatge serà: "`La consulta no ho produït resultats`".
8. Si l'opció és 4, demanar a l'usuari que introdueixi una data en format any-mes-dia i utilitzar el procediment `cerca_per_data` de l'ex.6 per a visualitzar la informació del partit jugat en la data indicada.
9. Si l'opció és 5, es surt del programa.
10. Qualsevol altra opció, escriure el missatge: `Opció no permesa`.
11. Repetir els passos 3 a 10, fins que l'opció del menú escollida sigui la 5.

Exercici 8 (1 punt)
Es vol implementar una app de participació ciutadana que permeti recollir dades sobre la quantitat i tipus de bolets que hi ha en els boscos d'un determinat territori. Com a part d'aquest projecte, se'ns demana definir la classe `CistellDeBolets`, que tindrà com a atributs un número enter amb la quantitat de bolets que hi ha al nostre cistell (per exemple, `4`) i una llista amb els tipus de bolets diferents que conté (per exemple, `["Rovellons", "Ceps", "Rossinyols"]`).

Definiu les següents funcions membre:

- Inicialització de la classe.
- `__str__` per tal de poder fer print's dels objectes de tipus `CistellDeBolets` amb el següent format:
  `"He caçat <quantitat> bolets: <tipus_bolet_1>, <tipus_bolet_2>,..."`
- `__add__` per tal de poder sumar dos cistells. La funció ha de retornar un nou cistell que contingui una llista amb els tipus de bolets que hi havia a les dues llistes (sense repetits) i la suma de les quantitats.
- `AfegirBolets(n, tipus)` que ens permetrà afegir `n` bolets del tipus `tipus` al cistell. La quantitat afegida es sumarà al total. El tipus només s'afegirà si són d'un tipus nou.
- `BuscaDolents(dolents)` que rebrà una llista de tipus de bolets suposadament dolents i retornarà una altra llista amb els tipus de bolets dolents que s'han trobat al propi cistell.

Exercici 9 (1 punt)

Fer una funció que es digui `map_filter` i que rebi com a paràmetre una llista de cadenes de text. La funció ha de retornar una altra llista que contingui només les cadenes de més de 3 caràcters i les ha de convertir a majúscules. Com a exemple, si la funció rep la llista `["a", "bb", "ccc", "dddd", "eeeee"]`, ha retornar la llista `["DDDD", "EEEEE"]`.

Notes: Cal utilitzar les funcions `map` i `filter` per donar l'exercici com a correcte.
Per a passar una cadena a majúscules podeu utilitzar el mètode `<cadena>.upper()`.

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 3 10:22:17 2020

@author: aitor
"""


# 1. Llegir fitxer per línies i tornar un diccionari

# jornada::tipus::data::equip::gols_marcats::gols_rebuts

def llegir_partits(nom_fitxer):
    filehandle=open(nom_fitxer,"r")
    partits={}
    for line in filehandle:
        valors=line[:-1].split('::')
        valors[4]=int(valors[4])
        valors[5]=int(valors[5])
        partits[int(valors[0])]=valors[1:]
    filehandle.close()
    return partits



# 2. Escriure fitxer a partir de diccionari.

def desar_partits(partits,nom_fitxer):
    filehandle=open(nom_fitxer,"w")
    jornades=list(partits.keys())
    jornades.sort()
    for x in jornades:
        dades=partits[x]
        dades[4]=str(dades[4])
        dades[5]=str(dades[5])
        dades=[str(x)]+dades
        filehandle.write('::'.join(dades)+'\n')
        #filehandle.write(str(x)+'::'+dades[0]+'::'+dades[1]+'::'+dades[2]+'::'+str(dades[3])+'
    filehandle.close()



# 3. Afegir element al diccionari.
def afegir_partit(partits,jornada,nom_fitxer):
    if jornada in partits:
        print("La jornada "+jornada+" ja conté dades")
    else:
        dades=[]
        dades.append(input("Introdueix la data en format any-mes-dia: "))
        dades.append(input("Introdueix local(L) o visitant(V): "))
        dades.append(input("Introdueix el nom de l'equip contrari: "))
        dades.append(int(input("Introdueix els gols marcats: ")))
        dades.append(int(input("Introdueix els gols rebuts: ")))
        partits[jornada]=dades
        desar_partits(partits,nom_fitxer)
```

```python
# 4. Obtenir histograma de gols rebuts.

def estadistica_gols_rebuts(partits):
    gols=[x[4] for x in partits.values()]
    histograma=[0 for x in range(10)]
    for x in gols:
        histograma[x]+=1
    return histograma


# 5. Obtenir jornades que tenen un determinat valor per una dada del partit.

def filtra_partits(partits,dada,valor):
    resultat=[]
    if dada=="LV":
        resultat=[k for k in partits if valor in partits[k]]
    elif dada=="EQ":
        resultat=[k for k in partits if valor in partits[k]]
    else:
        print("Error: Codi inexistent")
    return resultat

def filtra_partits2(partits,dada,valor):
    resultat=[]
    if dada=="LV":
        resultat=[k for k,v in partits.items() if v[1]==valor]
    elif dada=="EQ":
        resultat=[k for k,v in partits.items() if v[2]==valor]
    else:
        print("Error: Codi inexistent")
    return resultat

def filtra_partits3(partits,dada,valor):
    resultat=[]
    if dada=="LV":
        for k,v in partits.items():
            if v[1]==valor:
                resultat.append(k)
    elif dada=="EQ":
        for k,v in partits.items():
            if v[2]==valor:
                resultat.append(k)
    else:
        print("Error: Codi inexistent")
    return resultat


# 6. Obtenir dades del partit jugat en una determinada data.

def cerca_per_data(partits,data):
    res=[]
    for k,v in partits.items():
        if v[0]==data:
            res=[k]+v[1:]
    #res=[[k]+v[1:] for k,v in partits.items() if v[0]==data]
    if res==[]:
        print("Error: No hi ha cap partit en aquesta data")
    else:
        print("Jornada:",res[0]," - ",res[1],res[2],res[3],res[4])
```

```
################################################################################
def menu_principal():
    print("---- MENU ----")
    print("1.- Afegir partit")
    print("2.- Estadística de gols")
    print("3.- Consultar jornades")
    print("4.- Cercar partit per data")
    print("5.- Finalitzar")


# 7. Programa principal.

NOM_FITXER='campionat.txt'
try:
    D=llegir_partits(NOM_FITXER)
except FileNotFoundError:
    print("Fitxer no trobat")
except ValueError:
    print("El fitxer conté dades incorrectes")
else:
    opcio='0'
    while(opcio!='5'):
        menu_principal()
        opcio=input("Introdueix una opció del menú: ")
        if opcio=='1':
            jornada=int(input("Per quina jornada vols afegir un partit? "))
            while jornada <=0:
                print("Error: La jornada ha de ser un nombre positiu")
                jornada=int(input("Per quina jornada vols afegir un partit? "))
            afegir_partit(D,jornada,NOM_FITXER)
        elif opcio=='2':
            histo=estadistica_gols_rebuts(D)
            print("Estadística de gols rebuts: ")
            for x in range(10):
                print(str(x)+" - "+str(histo[x])+" partits")
        elif opcio=='3':
            dada=input("Per quina dada fols consultar, 'LV' o 'EQ'? ")
            valor=input("Quin és el valor a consultar? ")
            res=filtra_partits(D,dada,valor)
            if res==[]:
                print("La consulta no ha produït resultats")
            else:
                text="Resultat de la consulta: "
                for x in res:
                    text+=str(x)+','
                print(text[:-1])
        elif opcio=='4':
            data=input("Introdueix data en format any-mes-dia: ")
            cerca_per_data(D,data)
        elif opcio=='5':
            continue
        else:
            print("Opció no permesa")



################################################################################
```

```python
#8. Classe CistellDeBolets

class  CistellDeBolets():
    def __init__(self,quantitat,tipus):
        self.quantitat=quantitat
        self.tipus=tipus

    def __str__(self):
        mis="He caçat "+ self.quantitat + " bolets: "
        mis += ", ".join(self.tipus)
        return(mis)

    def __add__(self,other):
        q=self.quantitat+other.quantitat
        t=list(set(self.tipus+other.tipus))
        res=CistellDeBolets(q,t)
        return res

    def AfegirBolets(self,n,tipus):
        self.quantitat+=n
        if tipus not in self.tipus:
            self.tipus.append(tipus)

    def BuscaDolents(self,dolents):
        return [x for x in dolents if x in self.tipus]

#   versió alternativa de __add__:
#    def __add__(self,other):
#        res=CistellDeBolets()
#        res.quantitat=self.quantitat+other.quantitat
#        res.tipus=self.tipus.copy()
#        for x in other.tipus:
#            if x not in res.tipus:
#                res.tipus.append(x)
#        return res


#10. Funció map_filter
def map_filter(llista):
    return map(lambda x: x.upper(), filter(lambda x: len(x)>3,llista))

# prova
res=map_filter(["a","bb","ccc","dddd","eeeee"])
print(list(res))


# sense lambda functions
def f1(x):
    return x.upper()

def f2(x):
    return len(x)>3

def map_filter2(llista):
    return map(f1, filter(f2,llista))

# prova
res=map_filter2(["a","bb","ccc","dddd","eeeee"])
print(list(res))
```