



EXERCICI PUNTUABLE INFORMÀTICA.

Grau en Enginyeria de Sistemes TIC.

11/12/2017.

Exercici 1. Suposant que el següent script està contingut en `exercici.py` i que executem `python exercici.py sortida.txt` soc un bon estudiant i avui aprovaré. Què s'escriu per pantalla i què s'escriu en el fitxer `sortida.txt` després de l'execució d'aquest script `exercici.py`? Què conté `sys.argv`?

```
import sys
import string
f=open(sys.argv[1],"w")
print "Open out", sys.argv[1]
s=0; l=[]; d={}; vow="aeiouAEIOU"
for cad in sys.argv[2:]:
    cadAuxiliar=""
    for let in cad:
        if let in string.digits:
            s+=int(let)
        elif let not in vow:
            cadAuxiliar+=let
        else:
            d[let]=d.get(let,0)+1
    l+=[cadAuxiliar]
l.sort()
```

```

i=0
while i<len(l):
    print l[i]
    i+=1
llista=[]
for k,v in d.items():
    llista.append((v,k))
for element in llista:
    f.write(element[1]+"-"+str(element[0])+"\n")
f.close()
```

Exercici 2. Completa els doctests que segueixen.

```
def niIdea(d):
    """
    >>> niIdea({'Hidrogen':'H', 'Liti': 'Li', 'Sodi':'Na', 'Potassi':'K'})
    #apartat 2.1
    >>> niIdea({'Rodi': 'Rh', 'Liti': 'Li', 'Plata': 'Ag', 'Sodi':'Na'})
    #apartat 2.2
    """
    for clau,valor in d.items():
        if len(valor)==1:
            del d[clau]
    return d
```

Exercici 3. Escriu el resultat de l'execució.

```
def f(s, d):
    for k in d.keys():
        d[k]=0
    for c in s:
        if c in d:
            d[c] += 1
        else:
            d[c] = 0
    return d

def addUp(d):
    result = 0
    for k in d:
        result += d[k]
    return result

if __name__=='__main__':
    d1 = {}
    d2 = d1
    d1 = f('abbc', d1)
    print addUp(d1)
    d2 = f('bbcaa', d2)
    print addUp(d2)
    print f('', {})
```

Exercici 4. Les eleccions 11D. Dissenyà òptimament una funció tal que donat un fitxer de text (`resultats.txt`) que conté els resultats de les eleccions, amb el format, per cada línia, el codi partit i numero vots, se us demana que dissenyeu òptimament la funció `calculaResultat`, tal que escrigui en un nou fitxer (`resultatsnous.txt`), línia per línia cada codi partit i el percentatge de vots que ha obtingut respecte al total. A continuació segueixen els exemples d'execució.

```
#exemple contingut resultats.txt
partit1 1620973
partit2 336375
partit3 366494
partit4 102870
partit5 734910
partit6 522209
partit7 29785
partit8 14390
partit9 1158
partit10 326
```

```
#exemple contingut resultatsnous.txt
partit1 43.4636639326%
partit2 9.01932972069%
partit3 9.82692003464%
partit4 2.75828598548%
partit5 19.7053752658%
partit6 14.0021557907%
partit7 0.798634665866%
partit8 0.38584364082%
partit9 0.0310498218255%
partit10 0.00874114155019%
```

Exercici 5. La matriu matemàtica complexa. Dissenyeu òptimament una funció, tal que, donada una llista de llistes amb valors enters, corresponent a una matriu matemàtica de qualsevol ordre, retorni True, si hi ha almenys 2 files a la matriu amb almenys 3 valors igual a 8 (Vegeu els exemples). Expliqueu on apliqueu esquema/es de cerca/recorregut. Nota: No es valorarà una solució que no funcioni per matrius de qualsevol ordre.

Matriu a) Hauria de retornar True

```
8 1 8 1 1 8
8 8 8 4 2 8
1 -3 4 8 3 1
1 1 1 8 8 8
```

Matriu b) Hauria de retornar False

```
8 8 8
1 0 -1
1 1 2
```

```
def almenys3filesamb8(m):
    """
    retorna True si hi ha almenys 2 files a la matriu amb almenys 3 valors igual a 8 i False en cas contrari
    >>> almenys3filesamb8([[8, 1, 8, 1, 1, 8], [8, 8, 8, 4, 2, 8], [1, -3, 4, 8, 3, 1], [1, 1, 1, 8, 8, 8]])
    True
    >>> almenys3filesamb8([[8, 8, 8], [1, 0, -1], [1, 1, 2]])
    False
    """
```

Exercici 6. Detecció d'usuaris duplicats. Amb l'objectiu d'esbrinar si hi ha usuaris duplicats en dues xarxes socials, se us demana, que, donada la informació provinent de dues xarxes socials, retornem el llistat d'usuaris que es troben en ambdues xarxes. Per simplificar, suposarem que d'un usuari emmagatzemarem el seu nick (únic per cada usuari) i la seva paraula d'accés (no encriptada). A continuació segueixen els doctests de la funció que heu de resoldre òptimament.

```
def dicciocopy(d1,d2):
    """
    >>> dicciocopy({'juliaRoberts':'jaja22','shakira':'idon24','brattPitt':'noende','angelJol':'pepeps'},
    {'brattPitt':'noende','acdc':'88arroba'})
    {'brattPitt': 'noende'}
    >>> dicciocopy({'bandarra': 'carxofa'}, {})
    {}
    """
```

Exercici 7. El traductor. Ens passen un fitxer de text de nom *traduccions.txt* amb el següent format: El fitxer consisteix d'una paraula anglesa, seguida de qualsevol nombre de línies de la forma *xx=translation*, on *xx* correspon a lletres estandarditzades corresponents a codis de llenguatge, tal com, *de* per alemany, *cat* per català o *fr* per francès. Un exemple de fitxer seria el que segueix.

```
Cancel
de=Abbrechen
es=Cancelar
fr=Annuler
close
de=Schließen
cat=Tancar
fr=Fermer
OK
fr=Approuver
Open
de=Öffnen
es=Abrir
fr=Ouvrir
Today
cat=avui
```

Aquest fitxer ens diu, per exemple que la paraula anglesa *Cancel* s'hauria de visualitzar en alemany com a *Abbrechen*. En cas que no hi hagi idioma de traducció per a una paraula, es deixa la paraula anglesa. En cas que la paraula anglesa no es trobi en el fitxer, la traducció és "NotF"

Suposeu que tots els caràcters són d'expanded Unicode i per tant, no hi haurà accents ni caràcters especials. Se us demana que implementeu òptimament en python la funció *translate*, tal que, donat un nom de fitxer, una frase correcta (en anglès), i un codi de traducció, retorni la frase traduïda.

Nota: Podeu utilitzar funcions addicionals per tal d'optimitzar el codi solució.

1. Expliqueu on apliqueu esquema/es de cerca/reCORREGUT.
2. Expliqueu quina eina de gestió de dades òptima utilitzeu.

```
def translate(nom,frase,codi):
    """
    >>> translate("traduccions.txt","Cancel OK Open close","fr")
    'Annuler Approuver Ouvrir Fermer'
    >>> translate("traduccions.txt","Cancel all Open close Today","fr")
    'Annuler NotF Ouvrir Fermer Today'
    """
```