



Prova final d'INFORMÀTICA

25 de gener de 2013

Enginyeria de Sistemes TIC

90 MINUTS

COGNOMS:

NOM:

GRUP de LAB:

Exercici 1 [5 punts]. Imagineu que tenim una llista de contactes guardada a la targeta SIM. Cada contacte es representa amb una tupla amb el següent format: (nom/etiqueta, telèfon). L'etiqueta ens indica si el número de telèfon és de casa, el mòbil, del treball, etc... Per simplificar el problema suposarem que només hi ha dos tipus d'etiqueta: c que indica casa i m que indica mòbil. Per tant, si un contacte té número de casa i número de mòbil, es guardaran dues tuples a la llista.

En l'agenda dels mòbils, un contacte pot tenir diferents números de telèfon sense la necessitat de repetir el contacte guardat. Per tal efecte, dissenyeu una funció anomenada `conv_ic` tal que donada una llista de contactes, retorni un diccionari on el format de cada contacte serà el següent: `nom:[(etiqueta_completa, telèfon), (etiqueta_completa, telèfon)]` Un exemple d'execució seria el següent:

```
>>> l=[("Alba/c", "938881122")]
>>> l+= [("Alba/m", "615555555")]
>>> l+= [("Joan/c", "938771234")]
>>> conv_dic(l)
{'Alba': [('casa', '938881122'), ('mobil', '615555555')],
 'Joan': [('casa', '938771234')]}
```

SOLUCIÓ:

```
def conv_dic(l):
    d={}
    for x in l:
        nom=x[0][:-2]
        lloc=x[0][len(nom):]
        if d.has_key(nom):
            if lloc==" /c":
                d[nom].append(("casa",x[1]))
            else:
                d[nom].append(("mobil",x[1]))
        else:
            if lloc==" /c":
                d[nom]=[("casa",x[1])]
            else:
                d[nom]=[("mobil",x[1])]
    return d
```

Exercici 2 [3 punts]. Donada una matriu (llista de llistes) corresponent a una imatge d'una matrícula, dissenyeu una funció anomenada *separa* que retorni una matriu corresponent al primer dígit (el de més a l'esquerra). Supposeu que la primera columna (com a mínim) serà blanca. Recordeu que un píxel negre val 0 i un de blanc 255.

SOLUCIÓ:

En aquesta possible solució es fan servir dues funcions auxiliars perquè quedin clars els passos a seguir per resoldre el problema, però es podria resoldre d'una forma molt més compacte.

```
def columna_blanca(matriu,j):
    """
    Donada una matriu corresponent a la foto i un index j que
    correspon a la columna que volem mirar, retorna True si
    aquesta columna es de color blanc i False en cas contrari
    >>> columna_blanca([[255,0,255,255],[0,255,255,0],[0,0,0,0]],2)
    False
    >>> columna_blanca([[255,0,255,255],[0,255,255,0],[0,0,255,0]],2)
    True
    """
    i=0
    while i<len(matriu):
        x=matriu[i][j]
        if x==0:
            return False
        i+=1
    return True

def tallar_matriu(matriu,principi,final):
    """
    Donada una matriu i dos index, retorna la matriu retallada
    horitzontalment des de la columna principi fins la final.
    >>> tallar_matriu([[255,0,255,255],[0,255,255,0],[0,0,0,0]],1,3)
    [[0, 255, 255], [255, 255, 0], [0, 0, 0]]
    >>> tallar_matriu([[255,0,255,255],[0,255,255,0],[0,0,0,0]],0,2)
    [[255, 0, 255], [0, 255, 255], [0, 0, 0]]
    """
    numero=[]
    for fila in matriu:
        fila_nova=fila[principi:final+1]
        numero.append(fila_nova)
    return numero

def separa(matriu):
    """
    Donada una matriu corresponent a una imatge d'una matrícula,
    retorna la matriu corresponent al primer dígit.
    >>> separa([[255,0,255,255],[255,255,255,0],[255,0,255,0]])
    [[0], [255], [0]]
    >>> separa([[255,255,0,255],[255,255,0,255],[255,0,0,255]])
    [[255, 0], [255, 0], [0, 0]]
    """
    principi=False
    i=0
```

```

while principi==False and i<len(matriu[0]):
    if columna_blanca(matriu,i)==False:
        principi=True
    i+=1
j=i
final=False
while final==False and j<len(matriu[0]):
    if columna_blanca(matriu,j)==True:
        final=True
    j+=1

return tallar_matriu(matriu,i-1,j-2)

```

Exercici 3 [2 punts]. Seguiu el flux d'execució del programa següent i determineu la sortida:

Codi

Sortida

```

def a(x):
    print [0]*x

def b(x):
    abc="abcdefghi"
    return [abc[i] for i in range(x)]

def funcio(a,b="None"):
    if b==None:
        l=[[y for y in a(3)] for z in range(2)]
    elif b=="llista":
        l=[x for x in a(3)]
    else:
        l=[1,2,3]
    return l

l=funcio(b)
l2=a(4)
l2=funcio(b,l2)
print l
print l2

```

```

[0, 0, 0, 0]
[1, 2, 3]
[['a', 'b', 'c'], ['a', 'b', 'c']]

```