



NOM I COGNOMS:

DNI: 47823903-A

RESPONEU ELS APARTATS DEL PROBLEMA 1 EN AQUEST MATEIX FULL

Problema 1 [2.5 punts; tots els apartats tenen el mateix pes]

- 1.a) Dissenya una funció que donat un string que conté enters separats per comes, retorni el nombre d'enters que hi ha dins l'string. Per exemple, si la funció rep '12, 53, 6, 1, 0, -13', hauria de retornar 6.

```
def num-enters(s):
    p=s.split(',')
    return len(p)
```

2,5

- 1.b) Suposa que hem desat al fitxer *examen.py* la següent funció:

```
def misteri(d, a):
    for e in a:
        codi = e[0]
        if codi in d:
            d[codi][1] = d[codi][1] + e[2]
        else:
            d[codi] = e[1:]
```

Indica quins seran els valors de les variables *a* i *dicc* després d'executar les següents instruccions:

```
>>> import examen
>>> dicc = {'A1353': ['samarreta', 10], 'B322': ['bermuda', 3]}
>>> a = [['A1353', 'samarreta', 10], ['C335', 'camisa', 12]]
>>> examen.misteri(dicc, a)
```

a = [['A1353', 'samarreta', 10], ['C335', 'camisa', 12]]

dicc = {'A1353': ['samarreta', 20], 'B322': ['bermuda', 3], 'C335': ['camisa', 12]}

2,5



- 1.c) Donats els valors dels objectes *c*, *x* i *y* tal com s'indica, escriu una expressió a la línia de punts, on hi apareguin tots tres objectes i l'operació de concatenació, que permeti obtenir el nou valor de *c*.

```
>>> c = (3, 4)
>>> x = 2
>>> y = 5
>>> .....
>>> c
(3, 4, 10)
```

*c = c + (x*y,)*

2,5

- 1.d) La següent funció rep una llista d'strings i retorna una nova llista que conté els mateixos strings havent afegit al final de cadascun el sufix 'issim' i ordenada alfabèticament. Indica els errors que conté i torna a escriure-la correctament.

```
def issim(a):
    """
    >>> issim(['clar', 'alt', 'verd', 'curt'])
    ['altissim', 'clarissim', 'curtissim', 'verdissim']
    """
    b = []
    for e in a:
        s = e + 'issim'
        b = b + s
    b = b.sort()
    return b
```

def issim(a):

b = []

for e in a:

s = e + 'issim'

b.append(s)

b.sort()

return b

2,5



2359

Full núm.

Nombre total de fulls

Qualificació

40

Cognoms

Nom

Fonaments d'informàtica

Assinatura

DNI

Curs

60
Grup

07/01/14
Data

Problema 2

```
def analítica(l):  
    d={}  
    for i in range(len(l)):  
        for id in range(len(l[i])-1):  
            if l[i][i+1][0] in d:  
                d[l[i][i+1][0]].append(float(l[i][i+1][1]))  
            else:  
                d[l[i][i+1][0]] = [float(l[i][i+1][1])]  
    return d
```

Li

Problema 3

```
def num_curses(l, codi):  
    n=0  
    for e in l:  
        if e[0] == codi:  
            n=n+1  
            if n>=3:  
                return True  
    return False
```

```
def temps_curses(l, codi):
```

3002

Cognoms

Nom

Assignatura

d'informàtica

DNI

Curs

Grup

Data

60

07/01/14

Problema 3

def num_curses(l, codi):

n=0

~~for~~ e in l:

if e[0] == codi:

n=n+1

if n>=3:

return True

return False

def temps_cursor(l, codi):

~~for~~ e in l:

if e[0] == codi:

if e[2] < 10000:

pass

else:

return False

return True

def curses(nomfitxer, codi):

f=open(nomfitxer, 'r')

l_tot=[]

for linia in f:

linia=linia.strip()

l=linia.split('-:-')

e[2]=int(l[2][0:2]+l[2][3:5]+l[2][6:])

l_tot.append(l)

if num_curses(l_tot, codi) == True:

if temps_cursor(l_tot, codi) == True:

return True

f.close()

return False

quan troba
3 no passa

Problema 4

def permutacions (l1: list, l2: list):

d = {}

l = []

for e in l1:

d[e] = 0

for e in l2:

d[e[0]] = d[e[0]] + e[1]

for e in d:

l.append((e, d[e]))

return l

Aquesta funció no retornarà els tipus de
lletres en ordre!

def permutacions (l1: list, l2: list):

l = []

for e in l1:

l.append((e, 0))

for i in range(len(l1)):

for j in range(len(l1)):

if l1[i][0] == l1[j][0]:

l[i][1] = l[i][1] + l1[j][1]

return l