



NOM I COGNOMS:

DNI:

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 ns(s):  
    t = s.split(',')  
    n = len(t)  
    return n
```

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)
```

Valor de *a*: el mateix que al principi ja que aquesta variable no es modifica, només es recorre.
Valor de *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 = c + (x * y, )
>>> c
(3, 4, 10)
```

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.sort()
    return b
```

← falta coma

← error (no pot concatenar (lista i string).

← el mètode sort és una funció impura.

def issim(a):

```
b = []
for e in a:
    s = e + 'issim'
    b = b + [s]
b.sort()
return b
```

2.5



2210

Full núm.

Qualificació

Nombre total de fulls

10

Cognoms

Nom

Fonaments d'Informàtica

Assinatura

1r

10

7-1-14

DNI

Curs

Grup

Data

Problema 2

def analitica (llista):

dicc = {}

for e in llista: v

for i in range (1, len (e)): v

if e[i][0] not in dicc:

 dicc [e[i][0]] = e[i][1] v

else:

 dicc [e[i][0]].append (e[i][1]) v

return dicc



10

3044

Cognoms

Fonaments d'Informàtica

Nom

Assinatura

DNI

Curs

Grup

Data

1r

10

7-1-14

Problema 3

```
def curses (fitxer, codi):  
    f = open (fitxer, 'r')  
    n = 0  
    for linia in f:  
        linia = linia.strip()  
        l = linia.split ('-:-')  
        if l[0] == codi and l[2][2:] == '00':  
            n = n + 1  
        if n == 3:  
            f.close()  
            return True  
    f.close()  
    return False
```



Cognoms

Nom

Fonaments d'Informàtica

Assinatura

DNI

Curs

Grup

Data

1r

10

7-1-14

Problema 4

def pessuqueria (l1istatallats, l1istadia):

dicc = {}

lln = []

for e in l1istatallats:

dicc[e] = 0

lln = lln + [e]

for f in l1istadia:

dicc[f[0]] = dicc[f[0]] + f[1]

for i in range(len(lln)):

lln[i].append(dicc[lln[i]])

return lln