



Prova Final d'INFORMÀTICA

24/01/2014

Grau en Enginyeria de Sistemes TIC

2 HORE

COGNOMS:

NOM:

GRUP de LAB:

Exercici 1 [5 punts]. El verificador d'adreces de correu electrònic

[Apartat a] Es demana la implementació d'una funció de nom *check* que permeti detectar si una adreça de correu electrònic és correcta. Per simplificar, suposarem que una adreça de correu electrònic és correcta si té el format, *nom@servidor.extensio*, on, el *nom* pot ser qualsevol cadena de caràcters de com a mínim 3 lletres. El *servidor* només pot contenir lletres alfabètiques i la longitud del servidor ha de ser com a mínim de 5 caràcters. Una *extensió* es considera correcta si correspon a: *cat*, *com*, o bé *es*.

```
def check(email):  
    """  
    return True if email has correct format  
    >>> check('hola.hello@es')  
    False  
    >>> check('@.')  
    False  
    >>> check('pere@gmail.com')  
    True  
    >>> check('pere.gmail@com')  
    False  
    >>> check('ara@gmail22.com')  
    False  
    >>> check('pere@epsem.upc.es')  
    False  
    >>> check('jaon@gmail.com')  
    False  
    """
```

[Apartat b] Un cop dissenyada la funció, escriviu un petit script de nom *rastrejador.py*, tal que, donada una seqüència d'emails obtinguda per la línia de comandes, escrigui per pantalla el nombre d'emails incorrectes i escrigui en un fitxer de nom *rightEmails.txt* els emails correctes. El format d'escriptura a fitxer esperat correspon a una línia per adreça de correu correcta. A continuació segueix un exemple d'execució.

```
python rastrejador.py pere@epsem.upc.edu joan@gmail.com hola.hello@es pere@gmail.com ara@gmail22.com  
Detectats 3 emails incorrectes  
cat rightEmails.txt  
joan@gmail.com  
pere@gmail.com
```

Exercici 2 [3 punts]. La matrícula

Donada una matriu de pixels corresponent a una matrícula en blanc i negre, implementada com a llista de llistes on el valor 0 correspon al negre i el 255 al blanc, es demana que dissenyeu i implementeu les següents funcions.

[Apartat a] Funció *fBlack*, tal que reb com a paràmetre una matriu corresponent a una matrícula, i retorna True si hi ha una fila tota a negre en la matrícula i False en cas contrari. Completeu la documentació de la funció amb els doctests pertinents.

```
def fBlack(mat):
    """
    return True if there is a black row in mat
    >>> fBlack([[0,0,0,0],[0,255,255,0],[255,255,255,0],[0,255,255,255],[255,255,255,255]])
    True
    """
```

[Apartat b] Funció *cWhite*, tal que reb com a paràmetre una matriu corresponent a una matrícula, i retorna el nombre de columnes en blanc dins la matrícula. Completeu la documentació de la funció amb els doctests pertinents.

```
def cWhite(mat):
    """
    return the number of white columns in mat
    >>> cWhite([[0,255,255,255],[0,255,255,0],[255,255,255,0],[0,255,255,255],[255,255,255,255]])
    2
    """
```

Exercici 3 [2 punts]. Els nombres binaris

[Apartat a] Dissenyau i implementeu una funció *checkBinary* que rebi com a paràmetre una cadena composta de nombres binaris separats per espais en blanc, juntament amb una paraula que actua com a nombre binari a cercar, d'ara endavant, *binary target*, i mostri les posicions d'aquest *binary target* en la frase. Per exemple, si la cadena fos *001 100 111 101 010 001 001* i la *binary target* fos *001*, el resultat hauria de ser *0, 5, 6*, donat que *001* apareix en aquestes posicions en la cadena. La funció ha de retornar *False* si la *binary target* no apareix en la cadena.

```
def checkBinary(string,target):
    """
    return a list containing the positions of target in binary string. Return False if target not in string
    >>> checkBinary('001 100 111 101 010 001 001','001')
    [0, 5, 6]
    >>> checkBinary('001 100 111 101 010 001 001','1111')
    False
    >>> checkBinary('', '1111')
    False
    >>> checkBinary('001 100 111 101 010 001 001','')
    False
    >>> checkBinary('', '')
    False
    """
```

[Apartat b] Supposeu que esteu cercant *binary target* d'una llarga llista. Dissenyau i implementeu la funció *showAllTargets* tal que, donada una cadena de nombres binaris i una llista de *binary target*, mostri les posicions on es troben els *binary target* de la llista. Utilitzeu la funció *checkBinary* de l'apartat anterior. A continuació segueixen els doctests.

```
def showAllTargets(string,ltarget):
    """
    >>> showAllTargets('001 100 111 101 010 001 001',['001','0000'])
    Binary target 001 found in positions
    0 5 6
    Binary target 0000 not found
    >>> showAllTargets('001 100 111 101 010 001 001',[])
    Empty Binary targets
    >>> showAllTargets('', ['111'])
    Empty Binary string
    >>> showAllTargets('', [])
    Operation not allowed
    """
```