



Prova Final d'INFORMÀTICA

Grau en Enginyeria de Sistemes TIC

27/01/2015

COGNOMS:

NOM:

GRUP de LAB:

Exercici 1 [5 punts]. El filtrador de missatges corruptes.

[Apartat a] Es demana la implementació d'una funció de nom *missatgecorrupte(missatge, patro)* que permeti filtrar els missatges corruptes. Per filtrar un missatge corrupte se segueixen els següents passos:

- Els caràcters que no siguin lletres ni dígitos ni espais cal substituir-los per un espai en blanc
- Els caràcters que formin part del patro, cal substituir-los per *
- La resta de caràcters es mantenen
- Suposeu que patro conté únicament caràcters alfanumèrics

Adoneu-vos que hi pot haver missatges que no siguin corruptes, i el filtrat no tindrà efecte.

```
def missatgecorrupte(missatge, patro):  
    """  
    Retorna una còpia del missatge després d'aplicar filtratge  
>>> missatge_corrupte('Cor$r&r&^up&ted te**xt', 'jkqxyzJKQXYZ')  
'Cor  r up ted te *t'  
>>> missatge_corrupte('aqn eKvil vxiruzrs dzid ttyhis!', 'jkqxyzJKQXYZ')  
'a*n e*vil v*iru*rs d*id tt*his '  
    """
```

[Apartat b] Un cop dissenyada la funció, escriviu un petit script de nom *netejaMissatges.py*. Aquest script ha d'obtenir el patró i un nom de fitxer per la línia de comandes. Suposeu que aquest fitxer sempre existeix i no està buit. El fitxer conté un seguit de missatges, un per línia. Cal que l'script dissenyat escrigui en un nou fitxer que té el mateix nom que l'original afegint *rev* abans de l'extensió. En aquest nou fitxer hi ha d'haver els missatges corruptes filtrats, mantenint el format de cada missatge en una línia diferent. Finalment, ha de mostrar per pantalla el nombre de missatges corruptes. A continuació segueix un exemple de funcionament.

```
python netejaMissatges.py 'jkqxyzJKQXYZ' wrongMessages.txt  
2 missatges corruptes  
  
cat wrongMessages.txt  
Cor$r&r&^up&ted te**xt  
aqn eKvil vxiruzrs dzid ttyhis!  
No problem  
  
cat wrongMessagesrev.txt  
Cor  r up ted te *t  
a*n e*vil v*iru*rs d*id tt*his  
No problem
```

Exercici 2 [3 punts]. Les matrix.

[Apartat a] **El perímetre.** Donada una matriu de qualsevol ordre, representada com a llista de llistes, es demana que dissenyeu de manera òptima la funció que permeti retornar la suma dels elements que es troben en el perímetre de la matriu. Per exemple, en negreta es mostren els elements de la matriu que es consideren elements del perímetre de la matriu.

1 3 2

2 1 3

3 2 1

```
def perimeter(matrix):
    """
    retorna la suma dels valors del perímetre de matrix
    >>> perimeter([[1, 3, 2], [2, 1, 3], [3, 2, 1]])
    17
    >>> perimeter([[1, 2, 4, 3], [2, 3, 1, 3], [3, 1, 2, 3], [4, 2, 2, 4]])
    33
    """
```

[Apartat b] **El checkejador de sudokus.** Donat un sudoku de qualsevol ordre, representat com matriu quadrada, amb llistes de llistes, es demana que dissenyeu de manera òptima la funció checkSudoku, tal que donat un sudoku retorni True si aquest és correcte. Per simplificar, suposarem que un sudoku és correcte si mai coincideixin dos números iguals en cada línia horitzontal ni vertical.

```
def check_sudoku(sudoku):
    """
    retorna True si el sudoku es correcte
    >>> check_sudoku([[1, 3, 2], [2, 1, 3], [3, 2, 1]])
    True
    >>> check_sudoku([[1, 2, 4, 3], [2, 3, 1, 3], [3, 1, 2, 3], [4, 2, 2, 4]])
    False
    """
```

Exercici 3 [2 punts]. L'encriptador. Se'ns demana implementar una funcionalitat òptima que permeti gestionar l'enciptament i desenciptament de missatges que s'han d'enviar de manera privada entre un emissor i un receptor. En l'enciptació de missatges entre emissor i receptor es vol que cada lletra de l'abecedari se substitueixi per una altra lletra (a aquesta darrera l'anomenarem lletra enciptada). Suposarem que tant la lletra original com la enciptada han de correspondre a lletres minúscules.

Per exemple, es podria decidir que la lletra 'm' s'encipta amb una 'x', la lletra 'e' amb una 'z', la lletra 's' amb una 'f', la lletra 'a' com 'y' i la lletra 'g' com a 'w'. Per tant, amb aquesta codificació, el missatge 'message' s'enciptaria com a 'xzffyzw'

Fixeu-vos que cada lletra té una única lletra enciptada i que cada lletra enciptada pertany a una única lletra original.

[Apartat a] Justifiqueu com emmagatzemariéu de manera òptima la lletra original i la correspondència amb la lletra codificada.

[Apartat b] Dissenyeu una funció pura de nom *encipta*, tal que, donada una paraula, retorni la paraula enciptada.

[Apartat c] Dissenyeu una funció pura de nom *desencipta*, tal que, donada una paraula enciptada, retorni la paraula original.