



# Prova Final d'INFORMÀTICA

25/01/2018

Grau en Enginyeria de Sistemes TIC

COGNOMS:

NOM:

GRUP de LAB:

**NOTA IMPORTANT:** L'examen consta de 4 preguntes. Les preguntes s'han de resoldre en fulls separats.

## Exercici 1 [5 punts]. Query parameters.

Els Query parameters són una manera de gestionar la informació dins les URLs. Per exemple, quan cerquem a google informació sobre els query parameters, aquesta és la URL que genera automàticament Google.

```
http://www.google.com/search?sourceid=chrome&ie=UTF-8&q=query+parameters
```

Tal com es pot veure, Google emmagatzema la següent informació: que la cerca es va fer des del navegador chrome, que l'entrada de text era codificada amb format UTF-8, i que la cerca realitzada és 'query parameters.'

Més genèricament, els query parameters en les urls tenen el format que segueix.

```
http://website.com/?KEY1=VALUE1&KEY2=VALUE2...
```

Se us demana que:

**[Apartat 1.a]** Dissenyeu la funció *obtenirQueryParameters(url)* tal que, donada una URL, mostri per pantalla la informació dels query parameters. Per a l'exemple anterior, caldria que printés la informació que segueix. Adoneu-vos que el símbol + en les query parameters és indicador d'espai en blanc.

```
http://www.google.com/search?sourceid=chrome&ie=UTF-8&q=query+parameters
sourceid: chrome
ie: UTF-8
q: query parameters
```

O bé en un exemple de Facebook,

```
http://graph.facebook.com/search?q=watermelon&type=post
q: watermelon
type: post
```

Com que treballem a Facebook/Google, necessitem escriure aquestes funcions **òptimament**, o no podran ser evaluades. Podeu utilitzar les funcions predefinides de Python que necessitem. A continuació segueixen els doctests.

```

def obtenirQueryParameters(url):
    """
    >>> obtenirQueryParameters("http://www.google.com/search?sourceid=chrome&ie=UTF-8&q=query+parameters")
    sourceid: chrome
    ie: UTF-8
    q: query parameters
    >>> obtenirQueryParameters("http://graph.facebook.com/search?q=watermelon&type=post")
    q: watermelon
    type: post
    >>> obtenirQueryParameters("https://www.google.es/search?q=how+to+use+prezi&client=ubuntu&
channel=cs&aq=f&oq=how+to+use+prezi&aqs=chrome.0.57j65.18849&sourceid=chrome&ie=UTF-8")
    q: how to use prezi
    client: ubuntu
    channel: cs
    aq: f
    oq: how to use prezi
    aqs: chrome.0.57j65.18849
    sourceid: chrome
    ie: UTF-8
    """

```

[Apartat 1.b] Convertiu la funció anterior en *booleana* i afegiu que retorni **True** si la consulta realitzada té a veure amb *how to*. En els doctests anteriors, únicament el darrer test retornaria **True**. Els dos primers retornarien **False**. **Nota:** No és necessari que copieu de nou íntegrament el codi de la funció, afegiu indicació del codi que us caldria afegir i la seva ubicació.

[Apartat 1.c] Suposant que disposeu de la funció *obtenirQueryParameters* de l'Apartat 1.b, correctament implementada, escriviu un script tal que donats un conjunt de URLs que es troben en el fitxer *queries.txt* amb el format: una línia per URL, escrigui en el fitxer *estadistiques.txt*, quin percentatge d'aquestes URLs té a veure amb consultes de tipus *how to*.

```

#contingut fitxer queries.txt
https://www.google.es/search?q=how+to+use+prezi&client=ubuntu&channel=cs&aq=f&oq=how+to+us&
aqs=chrome.0.59j57j6213.2339&sourceid=chrome&ie=UTF-8
https://www.google.es/search?q=how+to+program+in+java&client=ubuntu&channel=cs&aq=f&oq=how+to+program&
aqs=chrome.0.59j57j60j6213.2098&sourceid=chrome&ie=UTF-8
https://www.google.es/search?q=pictures+of+nutshell&client=ubuntu&channel=cs&aq=f&oq=pictures+of+nutshell&
aqs=chrome.0.57.3219&sourceid=chrome&ie=UTF-8
https://www.google.es/search?client=ubuntu&channel=fs&q=selection+sort+algorithm&ie=utf-8&oe=utf-8&
gfe_rd=cr&dcr=0&ei=6rdUWvPsCZLY8geNwpPYBg

#el contingut del fitxer estadistiques.txt hauria de ser
50%

```

**Exercici 2 [1.5 punts]. Xarxes socials.** Suposant que teniu una xarxa social representada com un graf tal com mostra la Figura 1 esquerra.

Suposeu que dita xarxa s'ha representat amb un diccionari, de nom *xarxa*, on cada clau d'aquest serà el nick d'una persona, i el valor associat, una llista de nicks de les persones amb les quals és amic. Suposeu ara que cada persona de la xarxa social té un animal preferit. A tal efecte, es creu un nou diccionari, *animals*, on per cada clau corresponent al nick d'un usuari de la xarxa social, s'emmagatzema com a valor el seu animal preferit.

Direm que una persona de la xarxa social és un *hipster* si el seu animal preferit és diferent de tots els animals preferits dels seus amics. La Figura 1 dreta exemplifica aquesta situació. Per exemple, en la xarxa social, el conjunt de hipsters correspon a AngJolie, JuliaRoberts, Shakira i DavidBeckham. Se us demana que, implementeu **òptimament** la funció *hipster*, tal que, donats els dos diccionaris anteriors, retorni la llista de nicks d'aquells usuaris que són hipster. A continuació segueixen els doctests.

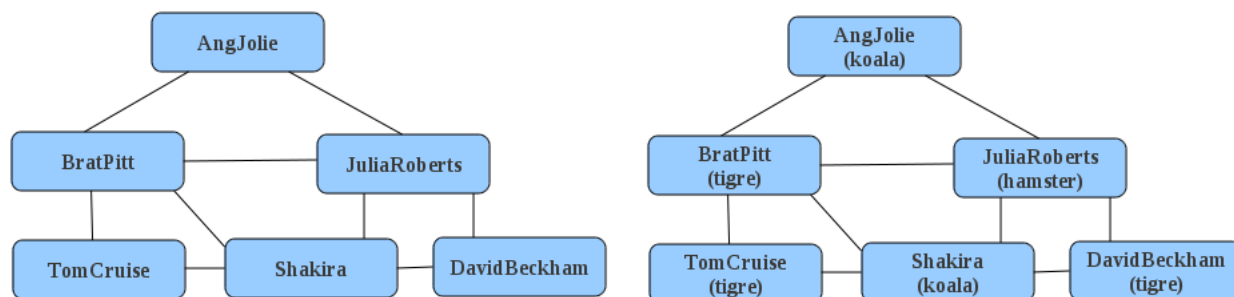


Figura 1: La xarxa social

```
def hipster(d,e):
    """
    >>> hipster({'AngJolie': ['BradPitt', 'JuliaRoberts'], 'JuliaRoberts': ['AngJolie',
    'BradPitt', 'Shakira', 'DavidBeckham'],'BradPitt': ['AngJolie', 'JuliaRoberts', 'TomCruise',
    'Shakira'], 'TomCruise': ['BradPitt', 'Shakira'], 'Shakira': ['TomCruise', 'BradPitt', 'JuliaRoberts',
    'DavidBeckham'], 'DavidBeckham': ['JuliaRoberts', 'Shakira']}, {'AngJolie': 'koala', 'JuliaRoberts':
    'hamster', 'BradPitt': 'tigre', 'TomCruise': 'tigre', 'Shakira': 'koala', 'DavidBeckham': 'tigre'})
    ['AngJolie', 'JuliaRoberts', 'Shakira', 'DavidBeckham']
    """
```

**Exercici 3 [1.5 punts]. Algorisme d'inserció.** Suposant una llista correctament ordenada en ordre ascendent, i un valor a inserir, se us demana el disseny **òptim** d'una funció **pura** tal que, donats aquests paràmetres, retorni una nova llista amb l'element inserit en la posició correcta, de manera que la llista continui estant ordenada ascendentment. La llista pot contenir valors repetits. Òbviament, la utilització de les funcions predefinides de Python sobre llistes suposarà una puntuació nul·la. Es pot utilitzar el mètode *len*, les llesques/slices i l'operació de concatenació de llistes. A continuació segueixen els doctests.

```
def insereixValor(valor,llista):
    """
    >>> insereixValor(4, [3, 5, 7, 8])
    [3, 4, 5, 7, 8]
    >>> insereixValor(18, [1, 9, 12])
    [1, 9, 12, 18]
    >>> insereixValor(-4, [1, 2, 3])
    [-4, 1, 2, 3]
    >>> insereixValor(2, [])
    [2]
    >>> insereixValor(45, [-1, 22, 33, 45, 104, 208])
    [-1, 22, 33, 45, 45, 104, 208]
    """
```

**Exercici 4 [2 punts]. Kerning (Interespai entre lletres).** El kerning en tipografia correspon a l'espai que s'afegeix entre lletres per tal de millorar les funcions visuals en el printatge d'aquestes lletres. Un exemple, és el que es mostra a la Figura 2. (Font: Wikipedia). En el primer cas no s'ha aplicat kerning (Kerning=0) i per tant, es printa una lletra al costat de l'altra. En la resta de casos s'ha aplicat un Kerning donat, i una lletra s'ha aproximat (o encavalcat) a l'altra. La Figura 2, mostra altres exemples amb Kerning=1 i Kerning=2.

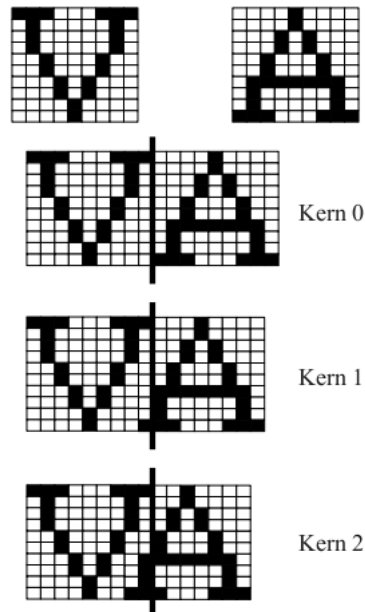


Figura 2: Exemple d'aplicació de kerning en el printatge de lletres

Per simplificar, suposarem que representarem les imatges de les lletres com a llistes bidimensionals de 0s i 1s indicant, per cada píxel de la imatge si és blanc (0) o negre (1). Per exemple, a continuació es mostra la representació de la lletra V i de la lletra A .

lletra V	lletra A
1 1 1 0 0 0 1 1 1	0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0	0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0	0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0	0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0	0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0	0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0	0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0	0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0	0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0	0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0	1 1 1 0 0 0 1 1 1

Se us demana que dissenyeu **òptimament** la funció *aplicarKerning*, tal que, donades dues matrius de 0s i 1s corresponents a lletres i un kerning, retorni la matriu resultant d'ajuntar les dues lletres amb el kerning indicat. Podeu suposar que l'alçada de les dues matrius serà la mateixa. No obstant, l'amplada de cada matriu pot diferir. Suposeu també que el kerning donat no és negatiu



i és inferior a l'amplada de qualsevol de les dues imatges. Noteu que si el kerning és zero, la imatge resultant serà l'amplada de les dues imatges posades juntes. A mesura que el valor del kerning augmenti, l'amplada de la imatge resultant disminuirà. Supposeu que les matrius donades seran sempre correctes i no serán mai buides.

**NOTA:** No està permès l'ús de llesques/slices ni la còpia de llistes. El seu ús comportarà una puntuació nul·la.

A continuació se us mostra el resultat d'aplicar diferents valors de kernings.

lletra V

```
1 1 1 0 0 0 1 1 1
0 1 0 0 0 0 0 1 0
0 1 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0
```

lletra A

```
0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0
0 0 1 0 0 0 1 0 0
0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 0
0 1 0 0 0 0 0 1 0
0 1 0 0 0 0 0 1 0
1 1 1 0 0 0 1 1 1
```

kerning 0

```
1 1 1 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 1 1
```

kerning 1

```
1 1 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1
```

kerning 2

```
1 1 1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 1 1 1 0 0 0 1 1 1
```

kerning 3

```
1 1 1 0 0 0 1 1 1 0 1 0 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0
0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 0
0 0 1 0 0 0 1 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 1 1 1 1 1 1 1 0
0 0 0 1 0 1 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 0 1 0 0 0 0 0 1 0
0 0 0 0 1 0 1 1 1 0 0 0 1 1 1
```