



Examen final d'INFORMÀTICA

20 de gener de 2012

Enginyeria de Sistemes TIC

120 MINUTS

Exercici 1 [2 punts].

1. Expliqueu breument i precisa què és l'àmbit de visibilitat d'una variable. Il·lustreu-ho amb un exemple.

SOLUCIÓ:

L'àmbit de visibilitat d'una variable és la regió sintàctica d'un programa en la que la variable es pot utilitzar. Per exemple, en la funció següent, la variable `w` té com a àmbit el cos de la funció i no és visible fora d'aquest cos.

```
def f1(a):  
    w = 23 + a  
    return w
```

2. Com sabeu, hi ha objectes que són mutables i altres que són immutables. Com afecta aquesta dualitat al pas de paràmetres en les funcions?

SOLUCIÓ:

Quan en cridar una funció es passa com argument un objecte immutable, la funció en qüestió no pot modificar-lo. En conseqüència no pot haver-hi efectes laterals. Quan l'argument és un objecte mutable, aquest pot ser modificat per la funció i es poden provocar efectes laterals.

3. El fitxer de text `dades.txt` conté 3 línies consecutives cadascuna de les quals té la cadena `'abcdxyz'`. Aleshores, quin és el resultat d'executar el següent codi i per què?

```
f = open('dades.txt', 'w')  
x = 'abcdxyz'  
for i in range(4):  
    f.write(x[:i] + '\n')  
f.close()
```

SOLUCIÓ:

En executar el programa, com el fitxer s'obre en mode escriptura el seu contingut desapareix. A partir d'aquí s'hi escriuen un seguit de línies que acaben deixant en el fitxer quatre línies amb el següent contingut:

- a) Una línia en blanc.
- b) Una línia amb la cadena `'a'`.
- c) Una línia amb la cadena `'ab'`.
- d) Una línia amb la cadena `'abc'`.

Exercici 2 [2 punts]. Sigui p un tuple que representa un polinomi d'una variable amb coeficients reals de manera que la component $p[i]$ correspon al coeficient de grau i del polinomi. Per exemple, el polinomi $f(x) = 3.2x^2 + 8.9$ es representaria amb el tuple $(8.9, 0, 3.2)$. Es demana que dissenyeu una funció tal que, donats un polinomi i un valor real retorni cert si el real és una arrel del polinomi. No oblideu el joc de proves basat en `doctest`.

SOLUCIÓ:

Que un real sigui arrel d'un polinomi significa que avaluant el polinomi per aquest valor s'obté zero. Així doncs, un joc de proves podria ser el següent:

```
def arrel(x,p):
    """
    >>> arrel(0.0, (0,0,2))
    True
    >>> arrel(2.0, (4,0,-1))
    True
    """
```

La solució consisteix en avaluar el polinomi per al valor x i comprovar si el resultat és zero. Es pot calcular de manera senzilla fent un recorregut:

```
r = p[0]
xn = float(x)
for c in p[1:]:
    r += c * xn
    xn *= x
return r
```

Exercici 3 [2 punts]. Una empresa que manufactura circuits electrònics disposa de dues màquines de muntatge de plaques. Al final del dia, cada màquina genera un diccionari que codifica el material que s'ha gastat. En aquests diccionaris la clau és una cadena amb la referència del component i el valor el nombre de components que s'han gastat. El següent diccionari ho exemplifica:

```
{'cond10mF': 23, 'res100M0': 321, 'ls74214': 2}
```

Es demana que dissenyeu una funció pura tal que donats dos diccionaris com l'anterior retorni un diccionari que conté la seva unió. Per exemple, si els dos diccionaris són:

```
{'cond10mF': 23, 'res100M0': 321, 'ls74214': 2}
{'res100M0': 100, 'ls74214': 20, 'res20': 70}
```

El resultat ha de ser el diccionari:

```
{'cond10mF': 23, 'res100M0': 421, 'ls74214': 22, 'res20': 70}
```

Acompanyeu la funció del joc de proves corresponent. Pareu atenció en el fet que es tracta de diccionaris.

SOLUCIÓ:

Per començar escriurem un joc de proves convenient. En aquest tindrem en compte que el resultat és un diccionari i els diccionaris no tenen un ordre ben definit. Per tant caldrà que la comprovació del resultat tingui en compte aquest detall.

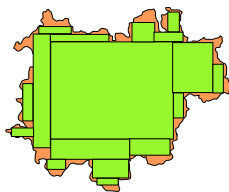
```
def unio(d1, d2):
    """
    >>> A = {'C0':2, 'C1':3, 'C2':4, 'C3':5}
    >>> B = {'C1':4, 'C2':4, 'C5':5}
    >>> R = unio(A,B)
    >>> R is A
    False
    >>> R is B
    False
    >>> R['C0']
    2
    >>> R['C1']
    7
    >>> R['C2']
    8
    >>> R['C3']
    5
    >>> R['C5']
    5
    >>> len(R)
    5
    """
```

A partir d'aquí per resoldre el problema primer clonem un dels diccionaris i, posteriorment, l'anem actualitzant recorrent l'altre:

```
r = d1.copy()
for k,v in d2.items():
    r[k] = r.get(k,0) + v
return r
```

Exercici 4 [4 punts]. Considereu el següent problema: es tenen un conjunt de coordenades obtingudes d'un GPS i es vol saber quantes corresponen a punts del Bages. Per resoldre aquest problema es fa la següent modelització.

Es representa el mapa d'una comarca és a través d'un conjunt de rectangles ortogonals de diferents mides que recobreixen aproximadament la superfície del mapa tal i com mostra la següent figura:



Cada rectangle ortogonal es pot representar com el parell de coordenades reals que corresponen als vèrtexs inferior esquerra i superior dreta. Si es coneixen aquest parell de punts, determinar si un punt arbitrari és dins o fora del rectangle és molt senzill. Comproveu-ho amb algun exemples fets a mà.

Usant aquest model és fàcil resoldre el problema original. Només cal representar un punt com un tuple de dues coordenades. Un rectangle com un tuple de dos punts: els dos primers representen les coordenades del punt inferior esquerra i els dos segons les del punt superior dreta. El mapa serà la llista dels rectangles que l'aproximen. Finalment, les coordenades del GPS són simplement una llista de punts. Per resoldre el problema procediu en el següent ordre:

1. Dissenyeu una funció pura `dins_rectangle(r,p)` tal que, donat un rectangle i un punt us indica si el punt és dins del rectangle o no.

SOLUCIÓ:

Simplement comprovem que les coordenades del punt es troben en l'interval definit pel rectangle:

```
def dins_rectangle(r, p):  
    v1, v2 = r  
    return v1[0] <= p[0] <= v2[0] and v1[1] <= p[1] <= v2[1]
```

2. Dissenyeu una funció pura `dins_mapa(m,p)` tal que, donat un mapa i un punt retorna cert sii el punt és dins del mapa (de l'aproximació del mapa).

SOLUCIÓ:

Només cal fer una cerca i comprovar si algun rectangle conté el punt:

```
def dins_mapa(m,p):  
    for r in m:  
        if dins_rectangle(r, p):  
            return True  
    return False
```

3. Dissenyeu la funció pura `quantas(m,l)` tal que donada una llista de punts i un mapa retorna quants punts de la llista són dins del mapa.

SOLUCIÓ:

Només cal recórrer la llista de punts i comptar els que són dins del mapa:

```
def quantas(m,l):  
    np = 0  
    for p in l:  
        if dins_mapa(m,p):  
            np += 1  
    return np
```

Per escurçar l'exercici, estalvieu-vos el joc de proves de les funcions.