



# EXERCICI PUNTUABLE INFORMÀTICA

16/10/2017

Grau en Enginyeria de Sistemes TIC

COGNOMS:

NOM:

GRUP de LAB:

**Exercici 1.** Escriviu exactament, què es mostra per pantalla, després d'executar cadascun dels fragments de codi que segueixen.

Apartat a)

```
omg = True
a = 0
b = 0
while omg:
    print "0"
    if a + b >= 24:
        omg = False
    a = a + 5
    b = b + 7
```

Apartat c)

```
def f(a):
    a = a + 5
    return a

b = 0
f(b)
print b, ",",
b = f(b)
print b
```

Apartat b)

```
a = 1
while a % 7 != 0:
    if a % 2 == 0:
        print "0",
    if a == 2:
        print "X"
    a = a + 1
```

Apartat d)

```
n = 10
i = 10
while i > 0:
    print i
    if i % 2 == 0:
        i = i / 2
    else:
        i = i + 1
```

Apartat e) Descriu també què fa la següent funció  $npi$ ,  $\forall n, n \in \epsilon, n > 0$ .

```
def npi(n):
    j=n/2
    while j>=1:
        if j**2==n:
            print j
        j--=1
```

**Exercici 2.** El calendari. Dissenyeu òptimament la funció *calendari* tal que, donat un número de l'1 al 12, corresponent al mes en curs, i l'any en curs, retorni quants dies té aquell mes.

Recordeu que, es consideren anys de traspàs, els anys en què les dues darreres xifres de l'any són múltiples de 4, excepte si aquestes xifres són 00. Aleshores, cal tenir en compte les dues primeres xifres de l'any. Si són múltiples de 4, l'any també serà de traspàs. Per exemple, el 1996 va ser de traspàs, perquè 96 és

múltiple de 4; el 1900, no fou de traspàs, perquè 19 no és múltiple de 4. Però el 2000, les dues darreres xifres també eren 00 i calia tenir en compte el 20. Com que és múltiple de 4, també va ser de traspàs. En resum: és de traspàs cada any múltiple de 4, excepte els múltiples de 100, que no ho són, i excepte els múltiples de 400, que sí que ho són. A continuació segueixen els exemples de funcionament.

```
i=1
while i<=12:
    print i,"2017:",calendari(i,2017),"1996:",calendari(i,1996),"1900:",
    print calendari(i,1900),"2000:",calendari(i,2000)
    i+=1
```

Resultat d'execució:

```
1 2017: 31 1996: 31 1900: 31 2000: 31
2 2017: 28 1996: 29 1900: 28 2000: 29
3 2017: 31 1996: 31 1900: 31 2000: 31
4 2017: 30 1996: 30 1900: 30 2000: 30
5 2017: 31 1996: 31 1900: 31 2000: 31
6 2017: 30 1996: 30 1900: 30 2000: 30
7 2017: 31 1996: 31 1900: 31 2000: 31
8 2017: 31 1996: 31 1900: 31 2000: 31
9 2017: 30 1996: 30 1900: 30 2000: 30
10 2017: 31 1996: 31 1900: 31 2000: 31
11 2017: 30 1996: 30 1900: 30 2000: 30
12 2017: 31 1996: 31 1900: 31 2000: 31
```

**Exercici 3.** Dissenyeu òptimament la funció *chequejaInput* tal que demani nombres a l'usuari fins que entri un nombre senar i divisible per 7. Quan l'hagi introduït, cal mostri per pantalla el missatge "Mission 7 odd accomplished" (acompanyat del número introduït senar i divisible per 7) i retorni el número d'intents realitzats. Chequegeu-ne el funcionament amb l'exemple que segueix.

```
>>> print str(chequejaInput())+" intents previs"
Entra nombre: 8
Entra nombre: 14
Entra nombre: 3
Entra nombre: 21
Number 21 Mission 7 odd accomplished
3 intents previs
```

**Exercici 4.** Dissenyeu òptimament la funció *triangleInvertitIticn* tal que, per qualsevol nombre positiu superior a 0, escrigui per cada línia, el número que rep com a paràmetre, decrementant en 2 unitats fins a arribar a l'1 o al 2. Aquest procediment s'ha de repetir per cada línia, començant cada línia pel valor 2 unitats inferior a l'anterior. El programa acaba quan escriu una línia amb un 1 o un 2. A continuació segueixen els exemples de funcionament.

```
>>> triangleInvertitIticn(20)
20 18 16 14 12 10 8 6 4 2
18 16 14 12 10 8 6 4 2
16 14 12 10 8 6 4 2
14 12 10 8 6 4 2
12 10 8 6 4 2
10 8 6 4 2
8 6 4 2
6 4 2
4 2
2

>>> triangleInvertitIticn(13)
13 11 9 7 5 3 1
11 9 7 5 3 1
9 7 5 3 1
7 5 3 1
5 3 1
3 1
1
```