

TD noté 2006 jeudi 30 novembre

Les deux exercices devront être imprimés à la fin du TD et rendus au chargé de TD. Il ne faut pas oublier de mentionner son nom en commentaires au début du programme.

1 Avec des fonctions uniquement

Le gouvernement désire ajouter un jour férié mais il voudrait le faire à une date éloignée des jours fériés existant. On suppose également que ce jour ne sera pas inséré entre Noël et le jour de l'an. On va donc calculer le nombre de jours qui sépare deux jours fériés dont voici la liste pour l'année 2007 :

Jour de l'an	1er janvier 2007
Lundi de Pâques	9 avril 2007
Fête du travail	1er mai 2007
Victoire de 1945	8 mai 2007
Ascension	17 mai 2007
Lundi de Pentecôte	4 juin 2007
Fête nationale	14 juillet 2007
Assomption	15 août 2007
Toussaint	1er novembre 2007
Armistice de 1918	11 novembre 2007
Noël	25 décembre 2007

On rappelle que l'année 2007 n'est pas une année bissextile et qu'en conséquence, le mois de février ne comporte que 28 jours.

1) Afin de simplifier la tâche, on cherche à attribuer un numéro de jour à chaque jour férié : l'année a 365 jours, pour le numéro du lundi de Pâques, soit 31 (mois de janvier) + 28 (février) + 31 (mars) + 9 = 89. La première question consiste à construire une fonction qui calcule le numéro d'une date étant donné un jour et un mois. Cette fonction prend comme entrée : (4 points)

- un numéro de jour
- un numéro de mois
- une liste de 12 nombres correspondant au nombre de jours dans chacun des douze mois de l'année

2) Si on définit la liste des jours fériés comme étant une liste de couples (jour, mois) triée par ordre chronologique, il est facile de convertir cette liste en une liste de nombres correspondant à leur numéro dans l'année. La fonction à écrire ici appelle la précédente et prend une liste de couples en entrée et retourne comme résultat une liste d'entiers. (4 points)

3) A partir de cette liste d'entiers, il est facile de calculer l'écart ou le nombre de jours qui séparent deux jours fériés. Il ne reste plus qu'à écrire une fonction qui retourne l'écart maximal entre deux jours fériés, ceux-ci étant définis par la liste de numéros définie par la question précédente. Un affichage du résultat permet de déterminer les deux jours fériés les plus éloignés l'un de l'autre. Quels sont-ils ? (4 points)

2 Programme équivalent avec des classes

Le programme précédent n'utilise pas de classe. L'objectif de ce second exercice est de le réécrire avec une classe.

1) Une fonction du programme précédent effectue la conversion entre un couple jour-mois et un numéro de jour. Les calculs sont faits avec le numéro mais le résultat désiré est une date : les numéros ne sont que des intermédiaires de calculs qui ne devraient pas apparaître aussi explicitement. La première question consiste à créer une classe `Date` : (2 points)

```
class Date :
    def __init__ (self, jour, mois) :
        ...
```

2) A cette classe, on ajoute une méthode qui retourne la conversion du couple jour-mois en un numéro de jour de l'année. (2 points)

3) On ajoute maintenant une méthode calculant le nombre de jours séparant deux dates (ou objet de type `Date` et non pas numéros). Cette méthode pourra par exemple s'appeler `difference`. (2 points)

4) Il ne reste plus qu'à compléter le programme pour obtenir les mêmes résultats que le programme de l'exercice 1. (2 points)

5) Avec ce programme, lors du calcul des écarts entre tous les jours fériés consécutifs, combien de fois effectuez-vous la conversion du couple jour-mois en numéro pour le second jour férié de l'année ? Est-ce le même nombre que pour le programme précédent (en toute logique, la réponse pour le premier programme est 1) ? (2 points)

6) La réponse à la question précédente vous suggère-t-elle une modification de ce second programme ? (2 points)

Correction

```
# coding: latin-1
#####
# exercice 1
#####

# question 1
def numero (jour, mois, duree = [31, 28, 31,30,31,30,31,31,30,31,30,31] ) :
    s = 0
    for i in range (0,mois-1) :
        s += duree [i]
    s += jour - 1
    return s+1

# question 2
def conversion_liste (li) :
    res = []
    for jour,mois in s : res.append ( numero (jour, mois))
    # pareil que
    # for i in range (0, len (s)) : res.append ( numero (s [i][0], s [i][1]))
    return res

def ecart (num) :
    res = []
```

```

for i in range (1, len (num)) :
    d = num [i] - num [i-1]
    res.append (d)
return res

s = [ (1,1), (9,4), (1,5), (8,5), (17,5), (4,6), (14,7), \
      (15,8), (1,11), (11,11), (25,12) ]
r = conversion_liste (s)
ec = ecart (r)

# question 3
pos = ec.index ( max (ec) )
print "position de l'écart le plus grand ", pos
print "jour ", s [pos], " --> ", s [pos+1]

#####
# exercice 2
#####

# question 4
class Date :
    def __init__ (self, jour, mois) :
        self.jour = jour
        self.mois = mois
        self.duree = [31, 28, 31,30,31,30,31,31,30,31,30,31]

    # question 5
    def numero (self) :
        s = 0
        for i in range (0,self.mois-1) :
            s += self.duree [i]
        s += self.jour - 1
        return s+1

    # question 6
    def difference (self, autre) :
        return self.numero () - autre.numero ()

def conversion_date (s) :
    res = []
    for jour,mois in s :
        res.append ( Date (jour, mois) )
    return res

def ecart_date (date) :
    ec = []
    for i in range (1, len (date)) :
        ec.append ( date [i].difference ( date [i-1] ) )
    return ec

# question 7
s = [ (1,1), (9,4), (1,5), (8,5), (17,5), (4,6), \
      (14,7), (15,8), (1,11), (11,11), (25,12) ]

r = conversion_date (s)
ec = ecart_date (r)
pos = ec.index ( max (ec) )

```

```

print "position de l'ecart le plus grand ", pos
print "jour ", s [pos], " --> ", s [pos+1]

# question 8
"""
La conversion en Date est faite une fois pour les dates (1,1) et (25,12)
et 2 fois pour les autres en effet, la méthode difference effectue
la conversion en numéros des dates self et autre
la fonction ecart_date calcule date [i].difference ( date [i-1] ) et
                                date [i+1].difference ( date [i] )
--> la date [i] est convertie 2 fois
"""

# question 9
"""
On peut par exemple stocker la conversion en numéro
dans le constructeur comme suit :
"""

class Date :
    def __init__ (self, jour, mois) :
        self.jour = jour
        self.mois = mois
        self.duree = [31, 28, 31,30,31,30,31,31,30,31,30,31]
        self.num = self.numero ()

    # question 5
    def numero (self) :
        s = 0
        for i in range (0,self.mois-1) :
            s += self.duree [i]
        s += self.jour - 1
        return s+1

    # question 6
    def difference (self, autre) :
        return self.num - autre.num

r = conversion_date (s)
ec = ecart_date (r)
pos = ec.index ( max (ec) )
print "position de l'écart le plus grand ", pos
print "jour ", s [pos], " --> ", s [pos+1]

```

On pourrait encore améliorer la dernière classe Date en créant un attribut statique pour l'attribut duree qui est identique pour toutes les instances de la classe Date.