

EXERCICE 1 (4 points)

Cet exercice porte sur les structures de données (listes, p-uplets et dictionnaires).

On dispose de la liste `jours` suivante et du dictionnaire `mois` suivant :

```
jours=["dimanche", "lundi", "mardi", "mercredi", "jeudi",  
"vendredi", "samedi"]
```

```
mois={1 : ("janvier",31) , 2 : ("février",28) , 3 : ("mars",31),  
4 : ("avril",30) , 5 : ("mai",31) , 6 : ("juin",30) ,  
7 : ("juillet",31) , 8 : ("aout",31) , 9 : ("septembre",30) ,  
10 : ("octobre",31) , 11 : ("novembre",30) ,  
12 : ("décembre",31) }
```

1.

- A partir de la liste `jours`, comment obtenir l'élément "lundi" ?
- On rappelle que l'opérateur % (« modulo ») renvoie le reste de la division entière (division euclidienne).

Exemple : $7\%3$ renvoie 1 qui est le reste de la division de 7 par 3 :
$$\begin{array}{r|l} 7 & 3 \\ -6 & 2 \\ \hline 1 & \end{array}$$

Que renvoie l'instruction `jours[18%7]` ?

- On rappelle que `jours.index[element]` renvoie l'indice de `element` dans la liste `jours` par exemple `jours.index["mercredi"]` renvoie 3. Le nom du jour actuel est stocké dans une variable `j` (par exemple : `j = "mardi"`). Recopier et compléter l'instruction suivante permettant d'obtenir le numéro du jour de la semaine `n` jours plus tard :
`numero_jour = (jours.index[...] + ...) % ...`

3.

- A partir du dictionnaire `mois`, comment obtenir le nombre de jours du mois de mars ?
- Le numéro du mois actuel est stocké dans une variable `numero_mois`, écrire le code permettant d'obtenir le nom du mois qu'il sera `x` mois plus tard à partir du dictionnaire `mois`.

Par exemple :

```
si numero_mois = 4 et x = 5, on doit obtenir "septembre"  
si numero_mois = 10 et x = 3, on doit obtenir "janvier"
```

4. On définit une date comme un tuple :

```
(nom_jour, numero_jour, numero_mois, annee) .
```

- a. Sachant que `date = ("samedi", 21, 10, 1995)`, que renvoie `mois[date[2]][1]` ?
- b. Ecrire une fonction `jour_suivant(date)` qui prend en paramètre une date sous forme de tuple et qui renvoie un tuple désignant la date du lendemain.

Par exemple :

```
jour_suivant( ("samedi", 21, 10, 1995) ) renvoie  
("dimanche", 22, 10, 1995)
```

```
jour_suivant( ("mardi", 31, 10, 1995) ) renvoie  
("mercredi", 1, 11, 1995)
```

On ne tient pas compte des années bissextiles et on considère que le mois de février comporte toujours 28 jours.