

PROGRAMMATION FONCTIONNELLE CAML
Durée : 1h30 avec QCM - Aucun document autorisé
QCM : 5 points ; Programmation : 15 points

1. Seconde après seconde

Écrire une fonction `secEnHeures` : `int -> int * int * int` qui prend un nombre de secondes `s` et le convertit en heures, minutes et secondes. Par exemple, 1000 secondes correspond à 16 minutes et 40 secondes.

```
#secEnHeures(1000) ; ;  
- : int * int * int = 0, 16, 40  
#secEnHeures(100000) ; ;  
- : int * int * int = 27, 46, 40 *)
```

2. Année après année

Une année est bissextile si

- l'année est divisible par 4 et non divisible par 100
OU
- l'année est divisible par 400

1. Écrire une fonction `bissextile` : `int -> bool` qui renvoie vrai si le paramètre d'entrée `n` est une année bissextile.

<pre>#bissextile(2016) ; ; - : bool = true</pre>	<pre>#bissextile(2017) ; ; - : bool = false</pre>
--	---

2. En déduire une fonction récursive `nextBi` : `int -> int` qui, à une année `n`, renvoie `n` si `n` est une année bissextile, et la prochaine année bissextile sinon.

<pre>#nextBi(2016) ; ; - : int = 2016</pre>	<pre>#nextBi(2017) ; ; - : int = 2020</pre>
---	---

3. Jour après jour

Tout cet exercice ne concerne que l'année 2017. Cette année étant non bissextile, elle comporte 365 jours. Rappelons qu'il y a 30 jours en Avril, Juin, Septembre et Octobre, 28 jours en Février et que tous les autres mois comptent 31 jours.

1. Écrire une fonction `nbJoursParMois : int -> int` qui a un entier, associe le nombre de jours dans le mois correspondant. Votre fonction devra être définie par filtrage.

```
#nbJoursParMois(7) ;;
- : int = 31
```

2. En déduire une fonction `dateValide : int * int -> bool` qui renvoie vrai si une date est valide.

```
#dateValide (15,6) ;;
- : bool = true
#dateValide(3,14) ;;
- : bool = false
```

3. Nous sommes le 16 Octobre. Écrire une fonction `datePasse : int * int -> bool` qui prend en paramètre une date (j, m) et renvoie vrai si cette date est déjà passée en 2017.

4. Écrire une fonction `age : int * int * int -> int` qui prend une date de naissance sous la forme d'un triplet d'entiers (j, m, a) et renvoie l'âge de cette personne.

```
age (20,7,2000) ;;
- : int = 17
#age(20,10,2000) ;;
- : int = 16
```

5. Écrire une fonction `nbJours : int * int -> int` qui calcule le nombre de jours qui se sont écoulés du 1er Janvier à la date (j, m) passée en paramètre. On pourra écrire une fonction récursive auxiliaire `nbJaux : int * int -> int` pour le calcul, la fonction `nbJours` ne faisant que vérifier la validité de la date et amorcer la récursivité.

```
#nbJours(16,10) ;;
- : int = 289
#nbJours(31,12) ;;
- : int = 365
```

6. Inversement, écrire une fonction `date : int -> int * int` qui prend en entrée un nombre de jours $j \leq 365$ et renvoie la date correspondante. (On pourra, suivant le modèle de la question précédente, introduire une fonction récursive auxiliaire pour le calcul).

```
#date(289) ;;
- : int * int = 16, 10
```



Question subsidiaire : Quelle est la marque de la voiture qui permet à Doc et Marty de voyager dans le temps dans les films "Retour vers le futur"?