Exercice 1 : Date

Représente une date par tuple (j, m, a) où j est le jour du mois m de l'année a.

* Écrire une fonction **antérieure** qui prend deux dates d1 et d2 en paramètres et retourne True si d1 est antérieure à d2.
* Écrire une fonction **age** qui prend deux dates d1 et d2 en paramètres et retourne le nombre d'années pleines entre d1 et d2
* Écrire une fonction **age\_mois** qui prend deux dates d1 et d2 en paramètres et retourne le nombre d'années pleines entre d1 et d2 et le nombre de mois.

Exercice 2 : Personne

On représente une personne par un tuple (nom, date) où nom est lui-même un tuple formé du prénom et du nom et date est également un tuple représentant la date de naissance comme dans l'exercice précédent.

Écrire une fonction qui prend une personne en paramètre et retourne un tuple formé du nom de famille et de l'année de naissance.

Exercice 3 : Durée

1. Écrire une fonction ***temps\_secondes***prenant en paramètres un tuple ***temps***contenant les trois valeurs d’une durée en heures, minutes et secondes et qui retourne le nombre total de secondes.   
   Écrire la ou les préconditions et teste la fonction à l’aide d’assertions.
2. Écrire une fonction temps\_h\_min\_s prenant en paramètres un entier secondes contenant une durée exprimée en secondes et qui convertis cette durée en heures minutes et secondes. La fonction renverra un tuple.

Écrire les préconditions et teste à l’aide d’assertions.

Exercice 4 : Liste personne

On veut créer un programme qui demande des noms, prénoms, date de naissance d’utilisateurs.

Les données de chaque individu seront stockées dans un p-uplet nommé (namedtuple) qu’on appellera Personneet qui aura les trois champs *nom*, *prenom¸ date\_naissance*.

Ces namedtuple seront stockés dans une liste (tableau) au fur et à mesure.

Une fois que la saisie s’arrête, les données seront affichées de la manière suivante :

Nom Prénom Date de naissance