

PROGRAMMATION FONCTIONNELLE CAML
Durée : 1h30 Aucun document autorisé

1. Année bissextile (expressions booléennes)

On rappelle qu'une année n est bissextile si n est multiple de 4, sauf si n est multiple de 100 à moins que n ne soit multiple de 400. Ainsi,

- 2019 n'est pas bissextile puisque 2019 n'est pas un multiple de 4 ;
- 2020 sera bissextile car elle est multiple de 4 et non multiple de 100 ;
- 3000 n'est pas bissextile car elle est multiple de 100 sans être multiple de 400 ;
- 2000 était bissextile car 2000 est multiple de 400.

Écrire une fonction `bissextile` : `int -> bool` déterminant si une année n est bissextile.

On définira localement les restes de n dans les divisions par 4, 100 et 400 et on n'utilisera pas de sélection. Tester votre fonction sur les exemples précédents.

2. Volume d'un ballon de rugby (fonctions réelles)

Le volume d'un ellipsoïde de révolution de hauteur h et de diamètre d est

$$V = \frac{\pi h d^2}{6}$$

1. Écrire une fonction `trunc1` : `float -> float` qui tronque un nombre réel à sa première décimale.
2. Écrire une fonction `volume` : `float * float -> float` qui donne en dm^3 (donc en litre), tronqué à la première décimale, le volume d'un ellipsoïde de révolution de hauteur h et de diamètre d exprimés en dm . Il n'y a donc aucune conversion à faire. On définira localement `pi=3.14`.

Certains prétendent qu'un ballon de rugby est un ellipsoïde de révolution de hauteur 2.9 dm et de diamètre 1.9 dm. Pour tester votre fonction, vous pourrez vérifier, que le volume d'un ballon de rugby est de 5.4 litres.

3. Demain est un autre jour (filtrage et chaînes de caractères)

On écrit une date comme un quadruplet constitué d'une chaîne de caractère (le jour de la semaine), du numéro du jour, du numéro du mois et du numéro de l'année.

La date d'aujourd'hui est donc (« Lundi », 14,10,2019) .

On souhaite écrire une fonction *demain*, qui en fonction d'une date, calcule la date du lendemain. On ne tiendra pas compte des années bissextiles et on considérera que février a 28 jours. On suppose que la date donnée est valide.

Commencer par écrire :

1. suivant : string -> string, définie par un filtrage exhaustif qui à une chaîne de caractère correspondant à un jour de la semaine, associe le nom du lendemain.

```
#suivant("Mardi") ; ;
- : string = Mercredi
```
2. apres : int * int * int -> int * int * int qui a un triplet (jour, mois, année) associe le triplet du jour d'après.

```
#apres(14,10,19) ; ;
- : int * int * int = 15, 10, 19
#apres(31,10,19) ; ;
- : int * int * int = 1, 11, 19
#apres(31,12,19) ; ;
- : int * int * int = 1, 1, 20
```
3. Écrire enfin la fonction *demain* : string * int * int * int -> string * int * int * int qui calcule la date du lendemain d'une date donnée.

```
#demain("Lundi",14,10,2019) ; ;
- : string * int * int * int = "Mardi", 15, 10, 2019
```

4. Plus grand chiffre d'un nombre (récursivité élémentaire)

Écrire une fonction récursive *pgChif* : int -> int donnant le plus grand chiffre d'un nombre entier. On commencera par écrire (ou mieux, on écrira localement) une fonction *max* calculant le maximum de deux entiers.

```
pgChif(12745) ; ;
- : int = 7
```

5. Nombres palindromes. (Récursivité... moins élémentaire)

Un nombre est palindrome si son écriture est symétrique (on peut le lire dans les deux sens), c'est par exemple le cas des nombres 2345432 ou 7447. On cherche à déterminer si un nombre est palindrome. Pour cela, on écrira successivement :

1. la fonction récursive `ajouTete` : `int * int -> int` telle que `ajouTete(n,c)` ajoute le chiffre `c` à gauche du nombre `n`.

```
ajouTete(7654,2) ; ;
- : int = 27654
```

2. la fonction récursive `miroir` : `int -> int`, utilisant `ajouTete`, qui renvoie le symétrique d'un entier `n`.

```
miroir(5764) ; ;
- : int = 4675
```

3. Écrire enfin, en utilisant les fonctions précédentes, la fonction `palindrome` : `int -> bool` qui détermine si un entier est palindrome.

```
palindrome(56456) ; ;
- : bool = false
palindrome(56465) ; ;
- : bool = true
```