

Nom :

Prénom :

Note : / 10

□ **Exercice 1** : *Compte à rebours*

1. Rappeler la définition d'une fonction récursive

Une fonction récursive est une fonction qui s'appelle elle-même.

2. Ecrire en C, une fonction itérative `compte_rebours` qui prend en argument un entier `n` ne renvoie rien et affiche les entiers de `n` à 0 puis "Partez !"

```

1 void compte_rebours(int n)
2 {
3     for (int i = n; i >= 0; i--)
4     {
5         printf("%d\n", i);
6     }
7     printf("Partez !\n");
8 }
```

3. Ecrire une version récursive de cette fonction qu'on appellera `compte_rebours_rec`

```

1 void compte_rebours_rec(int n)
2 {
3     if (n == 0)
4     {
5         printf("Partez !\n");
6     }
7     else
8     {
9         printf("%d\n", n);
10        compte_rebours_rec(n - 1);
11    }
12 }
```

□ **Exercice 2** : *un peu de OCaml*

On donne la définition de la fonction `mystere` en OCaml :

```

1 let rec mystere n =
2   if n < 10 then 1 else 1 + mystere (n/10)
```

1. Quel type est automatiquement inféré pour `n`? Pourquoi?

`n` est de type entier car la division `/` s'effectue forcément entre deux entiers (pour les flottants la division est `/.).`

2. Donner les résultats des appels suivants : `mystere 7`, `mystere 42`, `mystere 666`, `mystere 2023` en complétant le tableau ci-dessous

n	7	42	666	2023
mystere n	1	2	3	4

3. Proposer une spécification et un nom plus adapté pour cette fonction.

Cette fonction calcule le nombre de chiffres en base 10 de l'entier positif `n` donné en argument, on peut la renommer `nb_chiffres`