

Echauffement...

1. Lire `Slides-01-SyntaxeC`
2. *Nombres parfaits*. Un entier positif est parfait s'il est égal à la somme de ses diviseurs (y compris 1 et non compris lui-même).
Exemples : $6 = 1 + 2 + 3$, $28 = 1 + 2 + 4 + 7 + 14$.
Ecrire un programme qui saisit un entier positif N et affiche tous les nombres parfaits existant entre 0 et N.
3. *Factorielle*
 - Écrire un programme constitué d'au moins 2 fonctions (`main` et `factorielle`) qui calcule et affiche la factorielle d'un nombre. Attention, vous utiliserez `typedef` (cf. Cours) pour définir la fonction factorielle.

```
typedef short TypeEntier;  
TypeEntier factorielle(TypeEntier);
```
 - Écrire un programme qui calcule et affiche la factorielle pour des entiers de 1 à 15. Que remarquez-vous ? et pourquoi ?
 - Changer la définition de `TypeEntier` par un `unsigned long long` (plus "grand" type de base en C) et observer le changement des résultats pour le calcul de factorielle des entiers de 1 à 15. Les résultats sont-ils corrects ?
4. Lire `Slides-02-TypesDeDonnees`
5. Créer un compte sur <http://exercism.org> et suivez le parcours (*track*) sur le langage C. Faites un maximum d'exercices easy dans cet ordre : Difference Of Squares, Resistor Color (`enum`), Grains, Isogram (`strings`), hamming (`strings`, `pointers`), ... Attention, beaucoup d'exercices (même easy) nécessitent les notions suivantes : array, strings, enum, struct, pointers (cf. `Slides-02-TypesDeDonnees`)