Allons plus loin

1. Un nombre entier *d* est un diviseur de l’entier *n* si le reste de la division euclidienne de *n* par *d* est 0.

Par exemple 3 est un diviseur de 12 (12 = 4 3 + 0).

Ecrire une fonction Python nommée *diviseur* qui teste si un nombre entier *d* est un diviseur d’un nombre entier *n*. Dans ce langage, le reste de la division euclidienne de *n* par *d* s’écrit *n* % *d*.

1. Un nombre est dit **parfait** s'il est égal à la somme de ses diviseurs propres c’est-à-dire ses diviseurs autres que lui-même. C'est le cas, par exemple de 6 (1 + 2 + 3 = 6) ou de 28 (1 + 2 + 4 + 7 + 14 = 28).
2. Ecrire une fonction *parfait* testant si un nombre entier est parfait.
3. Ecrire un programme qui affiche la liste des nombres parfaits compris entre deux entiers saisis au clavier.
4. Deux nombres entiers *n* et *m* sont dits **amicaux** ssi la somme des diviseurs propres de chacun des deux entiers est égale à l’autre entier.

Par exemple, 220 et 284 sont amicaux (1 + 2 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284 et 1 + 2 + 4 + 71 + 142 = 220).

1. Ecrire une fonction *amis* testant si deux nombres entiers sont amicaux.
2. Ecrire un programme qui affiche la liste des nombres amicaux compris entre deux entiers saisis au clavier.
3. Un nombre est dit **premier** s'il admet exactement deux diviseurs : lui-même et l'unité.
4. Ecrire une fonction *premier* testant si un nombre entier est premier.
5. Ecrire un programme utilisant la fonction précédente pour afficher la liste des nombres premiers compris entre deux entiers saisis au clavier.
6. Tout nombre entier naturel supérieur ou égal à 2 admet une unique décomposition en un produit de facteurs premiers.
7. Ecrire la fonction *ppdp* retournant le Plus Petit Diviseur Premier d’un entier au moins égal à 2 reçu en paramètre d’entrée.
8. Ecrire la fonction *decomposer*retournant la décomposition en facteurs premiers d’un entier au moins égal à 2 reçu en paramètre d’entrée.
9. On souhaite simuler le lancer de deux dés et calculer la fréquence d’apparition de la somme des deux dés égale à 7.
10. Ecrire une fonction *somme* qui simule le lancer de deux dés à six faces et retourne la somme des deux résultats obtenus.
11. Ecrire une fonction *fréquence* de paramètre d’entrée *n* utilisant la fonction précédente qui retourne la fréquence d’apparition de la somme des deux dés égale à 7 pour la réalisation de *n* simulations, *n* étant un entier naturel.
12. Ecrire un programme en Python qui affiche la fréquence d’apparition de la somme des deux dés égale à 7 pour 100, 500, 1000, 5000, 10000, 50000, 100000, 500000 et 1000000 de lancers.