

DM 3

Informatique

Puzzle

Simon Dauguet
simon.dauguet@gmail.com

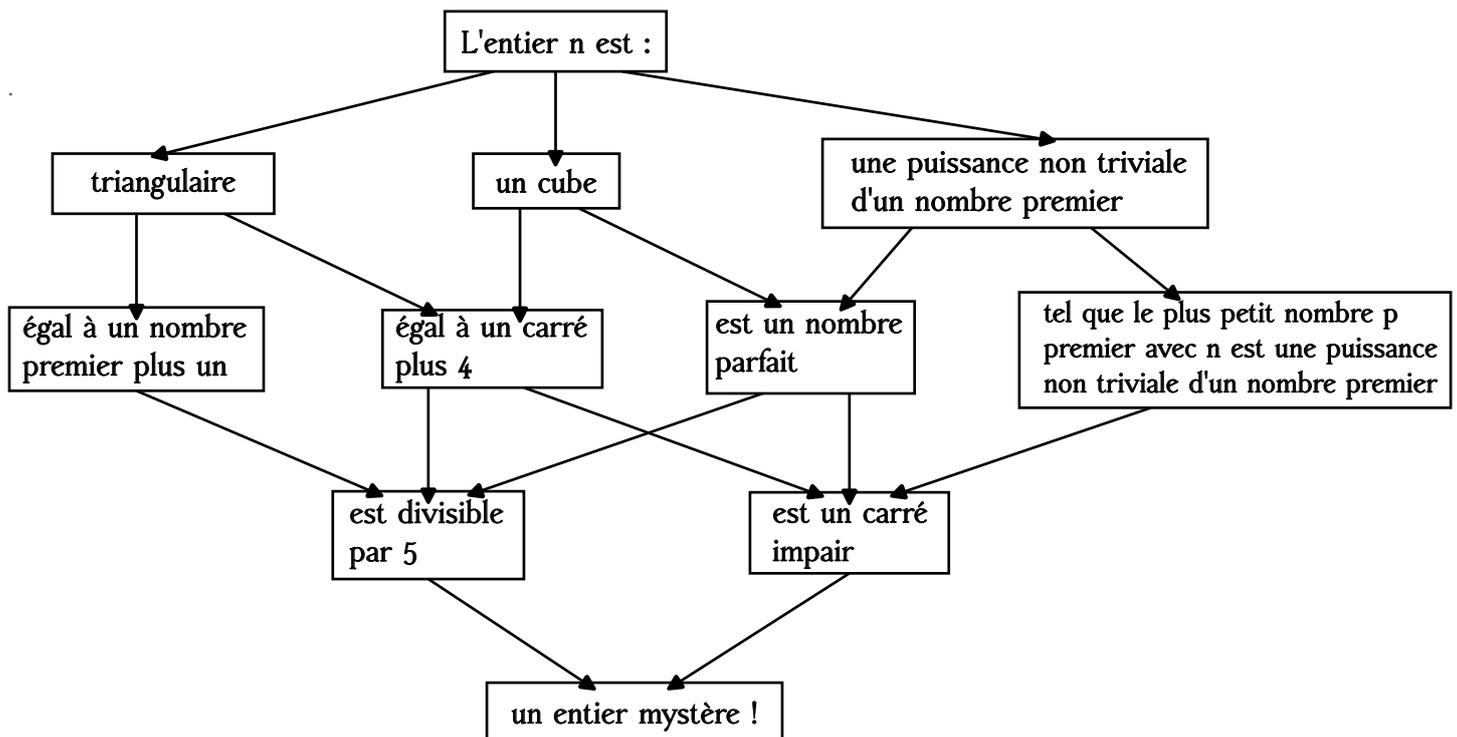
Pour le Vendredi 28 Février 2025

On renverra le code dans un fichier dont le nom devra être "DM3.Nom_Prénom.py" (attention, l'extension n'est pas toujours affiché par défaut sur les machines fonctionnant sous windows (c'est nul windows)).

On rappelle les quelques notions suivantes :

- Un entier parfait est un entier égal à la somme de ses diviseurs propres (exemple : $6 = 1 + 2 + 3$).
- Un nombre triangulaire est le nombre de pastille formant un triangle équilatéral. Autrement dit, un nombre qui peut s'écrire comme une somme de tous les entiers consécutifs. Autrement dit, $n \in \mathbb{N}$ est entier triangulaire, si $\exists p \in \mathbb{N}$ tel que $n = \sum_{k=1}^p k$.
- Deux entiers n et m sont premiers entre eux si leur pgcd (plus grand commun diviseur) est 1.
- On dira que n est une puissance non triviale d'un entier p si $\exists k \geq 2$ tel que $n = p^k$ et $p \geq 2$.

Le but de ce DM est de trouver les entiers qui vérifient le diagramme suivant :



-
1. Faire une fonction `square(n:int) -> list` qui renvoie la liste des carrés inférieurs ou égaux à n .
 2. Faire une fonction `cube(n:int) -> list` qui renvoie la liste des cubes inférieurs ou égaux à n .
 3. (a) Créer une fonction `div(n:int) -> list` qui renvoie la liste des diviseurs propres de n (donc différents de n).
(b) Faire une fonction récursive `pgcd(a:int, b:int) -> int` qui renvoie le pgcd de a et de b .
(c) Créer une fonction `minprem(n:int) -> int` qui renvoie le plus petit entier naturel différent de 1, premier à n .
(d) Créer une fonction `perfect(n:int) -> bool` qui renvoie `True` si n est entier parfait et `False` si non.
 4. Créer une fonction `triangle(n:int) -> bool` qui renvoie `True` si n est un entier triangulaire et `False` dans le cas contraire.
 5. (a) Créer une fonction `isprime(n:int) -> bool` qui renvoie `True` si n est premier et `False` dans le cas contraire.
(b) Créer une fonction `prime(n:int) -> list` qui renvoie la liste des nombres premiers inférieurs ou égaux à n .
(c) Créer une fonction `puissance(n:int) -> bool` qui renvoie `True` si n est une puissance non triviale d'un nombre premier, et `False` si non.
 6. Écrire enfin une fonction `chemin(n:int) -> list` renvoyant la liste de tous les entiers mystères dans $[[0, n]]$ (ne pas dépasser $n = 1000$ au maximum dans les tests).