

Interlude : nombres parfaits et nombres chanceux

Définitions :

- On appelle nombre premier tout entier naturel supérieur à 1 qui possède exactement deux diviseurs, lui-même et l'unité ;
- On appelle diviseur propre de n , un diviseur quelconque de n , n exclu ;
- un entier naturel est dit parfait s'il est égal à la somme de tous ses diviseurs propres ;
- les nombres a tels que : $(a + n + n^2)$ est premier pour tout n tel que $0 < n < (a - 1)$, sont appelés nombres chanceux.

Écrire un module (`parfait_chanceux_m.py`) définissant quatre fonctions : `somDiv`, `estParfait`, `estPremier`, `estChanceux` et un auto-test :

- la fonction `somDiv` retourne la somme des diviseurs propres de son argument ;
- les trois autres fonctions vérifient la propriété donnée par leur définition et retourne un booléen. Plus précisément, si par exemple la fonction `estPremier` vérifie que son argument est premier, elle retourne `True`, sinon elle retourne `False`.

Énoncés

La partie de test doit comporter quatre appels à la fonction `verif` permettant de tester

```
somDiv(12), estParfait(6), estPremier(31) et  
estChanceux(11).
```

Puis écrire le programme principal (`parfait_chanceux.py`) qui comporte :

- l'initialisation de deux tableaux : `parfaits` et `chanceux` ;
- une boucle de parcours de l'intervalle $[2, 1000]$ incluant les tests nécessaires pour remplir tableaux