Corrigé du DS n°1 d'arithmétique

Exercice 1

1. Le reste de la division euclidienne de 5^{2020} par 1337 s'obtient avec `5 ** 2020 % 1337` .
2. On fait des tests de 1 à $16 < \sqrt{280} < 17$.
    - $280 = 1 \times 280$
    - $280 = 2 \times 140$
    - $280 = 4 \times 70$
    - $280 = 5 \times 56$
    - $280 = 7 \times 40$
    - $280 = 8 \times 35$
    - $280 = 10 \times 28$
    - $280 = 14 \times 20$
    - Ainsi les diviseurs de 280 sont : $1, 2, 4, 5, 7, 8, 10, 14, 20, 28, 35, 40, 56, 70, 140, 280$.
3. $42! + 9$ est divisible par 9, donc il n'est pas premier.
4. La fonction `mystère` permet de calculer **le nombre de diviseurs d'un entier**. C'est une fonction écrite quasi comme la fonction `somme_diviseurs` vue en cours. Pour chaque diviseur `d` on incrémente `ans` de 1 qui était initialisé à 0.

```python
def mystère(n):
    ans = 0
    for d in range(1, n+1):
        if n % d == 0:
            ans = ans + 1
    return ans
```

Exercice 2

1. Calcul du PGCD avec la décomposition en facteurs premiers :
    - $144 = 2 \times 72 = 2^2 \times 36 = 2^3 \times 18 = 2^4 \times 3^2$.
    - $252 = 2 \times 126 = 2^2 \times 63 = 2^2 \times 3^2 \times 7$.
    - Ainsi $\mathrm{PGCD}(144, 252) = 2^2 \times 3^2 = 36$.
2.
    1. Tous les inscrits doivent être dans une équipe, et les équipes doivent être homogènes, donc le nombre d'équipe doit être un diviseur commun au nombre de garçons comme de filles. On en veut le maximum, ainsi le nombre d'équipes est $\mathrm{PGCD}(144, 252) = 36$.
    2. Chaque équipe est composée de :
        - $144 \div 36 = 4$ filles ;
        - $252 \div 36 = 7$ garçons.

Exercice 3

- $M_1 = 2^1 - 1 = 2 - 1 = 1$; n'est pas premier.
- $M_2 = 2^2 - 1 = 4 - 1 = 3$; est **premier**.

- $M_3 = 2^3 - 1 = 8 - 1 = 7$; est **premier**.
- $M_4 = 2^4 - 1 = 16 - 1 = 15 = 3 \times 5$; n'est pas premier.
- $M_5 = 2^5 - 1 = 32 - 1 = 31$; est **premier**.
- $M_6 = 2^6 - 1 = 64 - 1 = 63 = 3 \times 21$; n'est pas premier.
- $M_7 = 2^7 - 1 = 128 - 1 = 127$; est **premier**.
- $M_8 = 2^8 - 1 = 256 - 1 = 255$, divisible par 5 ; n'est pas premier.
- $M_9 = 2^9 - 1 = 512 - 1 = 511$, divisible par 7 ; n'est pas premier.
- $M_{10} = 2^{10} - 1 = 1024 - 1 = 1023$, divisible par 3 ; n'est pas premier.
- $M_{11} = 2^{11} - 1 = 2048 - 1 = 2047$, divisible par 23 ; n'est pas premier.

Remarques :

- Avant d'arriver à M_{11}, on aurait pu penser que M_n est premier si et seulement si n est premier. Mais c'est faux en considérant M_{11} qui est composé, alors que 11 est premier.
- On propose un script qui permet de vérifier davantage, et on peut prouver que M_n est premier que si n est premier ; sans la réciproque.
- L'exercice est sans fin, car aujourd'hui encore, la recherche des plus grands nombres premiers utilise ces nombres M_n qu'on appelle **nombres de Mersenne**. C'est un thème fécond en mathématiques pour de nombreux travaux.

```python
# avec la fonction is_prime vue en classe
for n in range(60):
    if is_prime(2**n - 1):
        print(f"M_{n} est premier.")
```

```
M_2 est premier.
M_3 est premier.
M_5 est premier.
M_7 est premier.
M_13 est premier.
M_17 est premier.
M_19 est premier.
M_31 est premier.
```

On constate que les indices sont bien premiers, mais il manque : $11, 23, 29, 37, 41, 43, 47, 53, 59$, ...

> **M_{67} est historiquement une première grande difficulté.**