

Nombre premier : Conjecture de Goldbach

Compétences visées

- **chercher**, expérimenter – en particulier à l'aide d'outils logiciels ;
- **modéliser**, faire une simulation, valider ou invalider un modèle ;
- **représenter**, choisir un cadre (numérique, algébrique, géométrique...), changer de registre ;
- **calculer**, appliquer des techniques et mettre en œuvre des algorithmes.

Ces compétences sont mises en œuvre dans le cadre de l'extrait du programme de 2^{nde} GT ci-dessous :

« Déterminer si un entier naturel est premier. » ou dans le cadre de l'extrait du programme de spécialité mathématiques en terminale S : « questionnement sur les nombres premiers »

Situation déclenchante

La conjecture de Goldbach affirme que tout nombre pair supérieur ou égal à 4 est la somme de 2 nombres premiers. Cette conjecture est-elle vraie ? Peut-on la tester sur les premiers entiers naturels ?

Problématique

Ecrire un programme qui permet de vérifier cette conjecture pour un entier naturel n donné. Le programme devra fournir tous les couples d'entiers premiers dont la somme fait n .

Fiche méthode

Proposition de résolution

A partir d'une liste de nombres premiers, on teste toutes les sommes possibles de manière à obtenir le nombre souhaité.

Ainsi, on crée trois fonctions dans ce programme (appelé aussi script) :

- Une fonction **premier(n)** qui permet de tester si un nombre est premier ou non. Cette fonction renvoie vrai ou faux.
- Une fonction **listepremiers(n)** qui permet de renvoyer la liste des nombres premiers inférieurs ou égaux à n.
- Une fonction **goldbach(n)** qui teste l'ensemble des sommes possibles de nombres premiers inférieurs ou égaux à n et qui renvoie tous les couples d'entiers premiers dont la somme fait n.

```
PYTHON SHELL
>>> from GOLDBACH import *
>>> premier(15)
False
>>> premier(17)
True
>>> listepremiers(21)
[2, 3, 5, 7, 11, 13, 17, 19]
>>> listepremiers(31)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31]
>>> |
Fns... | a A # | Outils | Éditer | Script
```

```
PYTHON SHELL
Configuration terminée.

>>> # L'exécution de GOLDBACH
>>> from GOLDBACH import *
>>> goldbach(24)
[[5, 19], [7, 17], [11, 13]]
>>> goldbach(34)
[[3, 31], [5, 29], [11, 23], [17, 17]]
>>> |
Fns... | a A # | Outils | Éditer | Script
```

Etapes de résolution

Fonction **premier(n)** renvoyant vrai ou faux si l'entier n est premier ou non.
L'instruction "for i in range(2,n)" permet de créer une boucle avec i variant de 2 à n - 1.
Pour avoir un programme plus efficace on aurait pu se limiter à $\text{floor}(\sqrt{n}) + 1$ dans la boucle.
L'instruction % permet de renvoyer le reste dans la division euclidienne.

```
ÉDITEUR : GOLDBACH
LIGNE DU SCRIPT 0001
def premier(n):_
    if n==1:
        return False
    for i in range(2,n):
        if n % i == 0:
            return False
    return True
Fns... | a A # | Outils | Exéc | Script
```

C'est un principe à retenir : On peut appeler une fonction (ici : la fonction **premier(n)**) à l'intérieur d'une autre fonction (ici : **listepremier(n)**). L'utilisation successive de fonctions en python rend le programme dans son ensemble plus lisible.

La fonction **listepremiers(n)** permet de retourner la liste des nombres premiers inférieurs ou égaux à n. L'instruction `listepremier.append(i)` permet de rajouter le nombre i à la liste primes.

```
ÉDITEUR : GOLDBACH
LIGNE DU SCRIPT 0022
def listepremiers(n):
    listepremier = []
    for i in range(2,n+1):
        if premier(i) == True:
            listepremier.append(i)
    return listepremier
Fns... | a A # | Outils | Exéc | Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Fiche méthode

Etapas de résolution

La fonction **goldbach(n)** permet de retourner les paires d'entiers premiers solution du problème. On fait appel à la fonction **listepremiers(n)** pour obtenir une liste de nombres premiers.

On utilise une double boucle pour générer toutes les sommes possibles de nombres premiers à partir de la liste obtenue grâce à la fonction **listepremiers(n)**.

La structure conditionnelle permet de tester si la somme est bien égale au nombre recherché et si la paire n'a pas déjà été sélectionnée.

```
ÉDITEUR : GOLDBACH
LIGNE DU SCRIPT 0029
def goldbach(n):
    *pairs = []
    *liste = listepremiers(n)
    *for premier1 in liste:
    *   for premier2 in liste:
    *       if premier1 + premier2 ==
    *           n and [premier1,premier2] n
    *           ot in pairs and [premier2,
    *               premier1] not in pairs:
    *           pairs.append([premier1,
    *               premier2])_
Fns... a A # Outils Exéc Script
```

```
ÉDITEUR : GOLDBACH
LIGNE DU SCRIPT 0031
*return (pairs)
Fns... a A # Outils Exéc Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

