Les nombres premiers.

1) Nombres premiers :

Définition 1 :

Un **entier** n **positif** est dit **premier** s'il possède **exactement deux diviseurs positifs distincts**, qui sont alors 1 et n.

Un **entier relatif** est alors premier s'il possède **exactement quatre diviseurs distincts**, qui sont donc $\{-n\,;-1\,;1\,;n\}$.

En conséquence, l'entier **1 n'est pas un nombre premier**, car il possède un seul diviseur positif qui est lui-même.

Si un entier n'est **pas premier**, il est dit **composé**.

Exemples : Voici la liste des nombres premiers inférieurs à 100 :

2	3	5	7	11	13	17	19	23	29	31	37	41	43
47	53	59	61	67	71	73	79	83	89	97			

Remarque : Pour un entier positif n, en démontrant que :

si d positif divise n, alors $d=1$ ou $d=n$

On démontre que n est premier.

Propriété 1 : **Critère d'arrêt**.

Soit un entier $n \geqslant 2$.

i) Alors n admet un diviseur premier.

ii) Le plus petit diviseur positif de n différent de 1 est un nombre premier.

iii) Si n n'est pas premier, ce plus petit diviseur p vérifie : $2 \leqslant p \leqslant \sqrt{n}$.

preuve :

Si n est premier, il admet un diviseur premier qui est lui-même.
Ce diviseur n est supérieur à 1 et il est le seul, donc il est également le plus petit.

Si n n'est pas premier, il possède un diviseur strictement compris entre 1 et n.
Notons alors p le plus petit diviseur de n strictement compris entre 1 et n et montrons que p est un nombre premier :

En effet, si un entier $d > 1$ est un entier qui divise p alors $d \leqslant p$.

Mais comme p divise n, alors d divise également n, donc d est plus grand que le plus petit des diviseurs de n, c'est-à-dire que $d \geqslant p$.

Ce qui donne finalement $d = p$, ce qui prouve que p est premier.

On peut donc écrire $n = p \times q$ avec p premier et aussi $p \leqslant q$ par minimalité de p.
Alors $p^2 \leqslant p\,q = n$ ce qui donne $p \leqslant \sqrt{n}$ par croissance de la fonction racine carrée.

Exemple : Montrons que 109 est un nombre premier.

On a $10 < \sqrt{109} < 11$.
On teste tous les nombres premiers strictement inférieurs à 11, soit : 2, 3, 5 et 7.
Des règles de divisibilités on déduit que 109 n'est divisible ni par 2, ni par 3, ni par 5.
En effectuant la division euclidienne de 109 par 7, on obtient : $109 = 7 \times 15 + 4$,
donc 109 n'est pas divisible par 7.

Finalement 109 est un nombre premier.

Exercice n°1 : Les entiers 481 et 487 sont-ils premiers ?

Exercice n°2 : Déterminer tous les diviseurs premiers de 504.

Algorithmique : Teste de primalité.

On cherche à déterminer sin un entier n est premier.

N'ayant pas à notre disposition la liste des nombres premiers,
on teste si n est divisible par 2 , puis on teste les diviseurs impairs suivants,
tant qu'ils sont inférieurs ou égaux à $\sqrt{n}$.

En version langage naturel :	En version Python avec une fonction :

d prend la valeur 2
si n est divisible par d alors :
afficher « n est divisible par d »
fin du programme
d prend la valeur d + 1
tant que $d \leqslant \sqrt{n}$ faire :
si n divisible par d alors :
afficher « n est divisible par d »
fin du programme
d prend la valeur d + 2
afficher n est premier
	`# fonction teste de primalité classique`
	`def primary_test(n):`
	`  div = 2`
	`  if n % div == 0:`
	`  print(n," est divisble par", div)`
	`  sys.exit()`
	`  div = div + 1`
	`  while div <= sqrt(n):`
	`  if n % div == 0:`
	`  print(n,"est divisble par", div)`
	`  sys.exit()`
	`  div = div + 2`
	`  print(n,"est premier")`
	
	`# programme principal`
	`import sys`
	`from math import sqrt`
	`entier = int(input("Donner un entier :"))`
	`primary_test(entier)`

Exemple : Avec l'algorithme précédent, on obtient successivement :

- 527 est divisible par 17
- 719 est premier
- 11 111 est divisible par 41
- 37 589 est premier.

Propriété 2 : L'ensemble des nombres premiers est infini.

Preuve : Raisonnons par l'absurde.

Supposons qu'il existe un nombre fini n de nombres premiers : p_1 , p_2 , … , p_n .
Alors l'entier $N = p_1 \times p_2 \times \ldots \times p_n + 1$ est supérieur ou égal à 2,
donc il possède un diviseur premier p_i d'après le critère d'arrêt.
Alors p_i divise N , et p_i divise $p_1 \times p_2 \times \ldots \times p_n$,
donc p_i divise 1 comme combinaison linéaire, ce qui est impossible.

Donc il ne peut pas y avoir un nombre fini d'entiers premiers.

2) Crible d'Ératosthène :

Un **multiple propre** de n est un multiple de n différent de n.

Pour dresser la liste des nombres premiers entre 2 et N, la méthode du crible d'Ératosthène consiste à :

- écrire la liste des entiers de 2 à N
- éliminer les multiples propres de 2, 3, …
- puis ceux du premier nombre non encore éliminé, et ainsi de suite.

Exemple : liste des nombres premiers de 2 à 150

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150

Les entiers éliminés (sur **fond bleu**) sont les **entiers composés**.
Les entiers restants (sur **fond jaune**) sont les **nombres premiers**.

Remarque :

1) Pour éliminer les multiples propres de 7, on peut commencer à 7^2 car les multiples inférieurs ont déjà été éliminés.

2) Il est possible de savoir à l'avance où s'arrêter. En effet, grâce au critère d'arrêt, tout entier composé n admet un diviseur premier $p \leqslant \sqrt{n}$.

Si $n \leqslant 150$, alors $\sqrt{n} \leqslant \sqrt{150}$, et comme $12 < \sqrt{150} < 13$ alors tout entier non premier sera éliminé en tant que multiple propre de 2, 3, 5, 7 et 11.

Algorithme : Liste des nombres premiers inférieurs à un entier donné.

On peut écrire l'algorithme suivant, qui nécessite quelques connaissances sur la manipulation des listes en Python.

```
# liste des nombres premiers de 2 à n
nombre = int(input("Donner un entier :"))
liste_nb_premiers = [k for k in range(2, nombre + 1)]
for nb in liste_nb_premiers:
    dernier = liste_nb_premiers[-1]
    liste_quotients = [liste_nb_premiers[i]\
                       for i in range(liste_nb_premiers.index(nb), len(liste_nb_premiers))\
                       if nb*liste_nb_premiers[i] <= dernier]
    for quotient in liste_quotients:
        liste_nb_premiers.remove(nb*quotient)
print(liste_nb_premiers)
```

On demande un entier **nombre**.
On créé la **liste** des entiers de 2 à **nombre**.
C'est notre réponse de départ, provisoire, et dans laquelle on va éliminer les nombres composés au fur et à mesure.
L'entier **nb** parcourt donc cette **liste** et à chaque étape :
On détermine tout d'abord la **liste des quotients** possibles.
Puis on élimine de la **liste** de nombres premiers tous les multiples correspondants.
On affiche la liste à la fin.

Remarques :

Le principe est de faire le moins de multiplications possible.
Lorsqu'on regarde un entier nb dans notre liste provisoire, on a besoin uniquement d'éliminer les multiples de nb par des entiers de cette liste, et en commençant par lui-même.

Donc le plus petit multiple possible est toujours nb^2 , car les multiples par des entiers inférieurs à nb ont déjà été éliminés.
Et le plus grand multiple possible ne doit pas dépasser le plus grand nombre de cette liste.

Voici le résultat pour 1 000 :

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131,
 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271,
 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433,
 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601,
 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769,
 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953,
 967, 971, 977, 983, 991, 997]
```

3) Divisibilité et nombres premiers :

Une conséquence du lemme de Gauss pour les nombres premiers est le résultat suivant, parfois appelé Lemme d'Euclide si je ne me trompe pas.

Propriété 3 : Lemme d'Euclide.

Un nombre premier divise un produit si et seulement s'il divise l'un des facteurs.

Si p est premier, alors : p divise ab $\Leftrightarrow$ p divise a ou p divise b

Autrement dit, les nombres premiers sont les nombres irréductibles pour la divisibilité.

preuve :

CN : Si p divise ab alors soit p divise a , soit il ne divise pas a ;
dans ce cas, p est un nombre premier qui ne divise pas a ,
donc p et a sont premiers entre eux.
Donc p divise b , d'après le lemme de Gauss.

CS : évidente

Propriétés 4 : Conséquences de ce qui précède.

1) Si p premier divise une puissance a^k alors p divise a , et donc p^k divise a^k .

2) Si p divise un produit de facteurs premiers,
alors p est l'un de ces facteurs premiers.

3) Soit $p_1 , p_2 , \dots , p_k$ des nombres premiers distincts,
et $\alpha_1 , \alpha_2 , \dots , \alpha_k$ des entiers naturels non nuls tels que :

Si $\forall i \in \{1,2,\dots,k\}$, $p_i^{\alpha_i}$ divise un entier n ,

alors $p_1^{\alpha_1} \times p_2^{\alpha_2} \times \dots \times p_k^{\alpha_k}$ divise l'entier n .

4) Théorème fondamental de l'arithmétique :

Propriété 5 : Théorème fondamentale de l'arithmétique.

Tout entier $n \geqslant 2$ peut se décomposer de manière unique (à l'ordre des facteurs près) en produit de facteurs premiers, donc sous une unique forme :

$n = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \ldots \times p_k^{\alpha_k}$ où les p_i sont des nombres premiers distincts, et les exposants α_i des entiers positifs ou nuls.

Exemple : La décomposition de 15 750 en facteurs premiers donne : $15\,750 = 2 \times 3^2 \times 5^3 \times 7$.

Pour décomposer un entier, on effectue des divisions successives par des nombres premiers dans l'ordre croissant.

Remarque :

Le théorème est encore valable pour $n = 1$.
Il faut choisir $k = 0$, car le produit d'aucun entier est par convention égal à 1.
On a bien $x^0 = 1$. On peut voir 1 comme le produit de la famille finie de nombres premiers, qu'est la famille vide.

preuve :

i) Démontrons l'existence de la décomposition par récurrence :

Initialisation :
$n = 2$ est bien un produit de facteur premier, étant lui-même premier.

Hypothèse de récurrence **forte** :
pour $k \geqslant 3$ fixé, on suppose que **tout entier** strictement inférieur à k est produit de facteurs premiers.

Hérédité : Démontrons que k est lui même produit de nombres premiers.
Soit p le plus petit entier supérieur à 1 divisant k.
Alors p est un nombre premier.
En effet, tout entier qui divise p divise k, donc par minimalité de p, il vaut p ou 1.
On peut alors écrire que $k = np$ avec $p > 1$ et donc $np = k > n$ ou $n < k$.
Alors par hypothèse, n est un produit de facteurs premiers, donc k également.

ii) Démontrons l'unicité intuitivement grâce au lemme d'Euclide :

Si p premier divise le produit ab alors il divise a ou b.

Si deux produits de nombres premiers sont égaux, on prend n'importe quel nombre premier p du premier produit. Il divise le premier produit, et donc le second.
D'après le lemme d'Euclide, il divise l'un des facteurs du second produit.
Mais ce sont des nombres premiers, donc p est nécessairement égal à l'un de ces facteurs. On peut alors simplifier l'égalité par p.
En continuant le procédé, on voit que les facteurs premiers des produits doivent coïncider...

Algorithme : Liste des facteurs premiers d'un entier donné.

On demande un entier **nombre**.
On crée une **liste** de diviseurs premiers vide.
Le premier diviseur **div** à tester est 2,
puis on teste les diviseurs impairs suivants
en appliquant le critère d'arrêt $\text{div} \leqslant \sqrt{nombre}$.

```
# liste des facteurs premiers d'un entier
from math import sqrt
nombre = int(input("Donner un entier :"))
liste_diviseurs = []
div = 2
pas = 1
while div <= sqrt(nombre):
    if nombre % div == 0:
        liste_diviseurs.append(div)
        nombre = nombre // div
    else:
        div = div + pas
        pas = 2
liste_diviseurs.append(nombre)
print(liste_diviseurs)
```

Si le **reste** de **nombre** par **div** est nul,
on ajoute **div** à la **liste** des diviseurs,
puis le **nouveau nombre** est le **quotient** de la division de **nombre** par **div**.

Sinon, on passe à 3, puis ensuite on avance de 2 en 2.

Le dernier nombre ne vérifiant par le critère d'arrêt est un nombre premier, donc il faut l'ajouter à la **liste** en fin de programme.

Exemples :

Avec 16 758 on obtient $liste = \{ 2 ; 3 ; 3 ; 7 ; 7 ; 19 \}$
Avec 87 616 on obtient $liste = \{ 2 ; 2 ; 2 ; 2 ; 2 ; 37 ; 37 \}$
Avec 77 986 545 on obtient $liste = \{ 3 ; 5 ; 7 ; 13 ; 19 ; 31 ; 97 \}$

Exercice n°3 : Déterminer la décomposition de 1 716 en produit de facteurs premiers.

Exercice n°4 :

1) Donner la décomposition en produits de facteurs premiers de 126 et 735.

2) En déduire leur pgcd et leur ppcm.

Exercice n°5 :

1) Soit un entier $n \geqslant 1$. Le nombre $n^2 - 1$ peut-il être premier ?

2) Soit un nombre premier $p \geqslant 5$.
En utilisant les congruences, montrer $p^2 - 1$ est divisible par 3.

Propriété 6 : Diviseurs d'un entier. Soit $n \geqslant 2$.

i) Si $n = p_1^{\alpha_1} \times p_2^{\alpha_2} \times \ldots \times p_k^{\alpha_k}$, alors tout diviseur de n peut s'écrire sous la forme :

$$m = p_1^{\beta_1} \times p_2^{\beta_2} \times \ldots \times p_k^{\beta_k} \text{ avec } 0 \leqslant \beta_i \leqslant \alpha_i \quad \forall i \in \{1, 2, \ldots, n\}.$$

ii) Le nombre de diviseurs positifs de n est : $d(n) = (\alpha_1+1)(\alpha_2+1)\ldots(\alpha_k+1)$.

iii) La somme de ces diviseurs est : $\sigma(n) = \dfrac{p_1^{(\alpha_1+1)} - 1}{p_1 - 1} \times \dfrac{p_2^{(\alpha_2+1)} - 1}{p_2 - 1} \times \ldots \times \dfrac{p_k^{(\alpha_k+1)} - 1}{p_k - 1}$.

iv) Le produit de ses diviseurs positifs est : $p(n) = n^{\frac{d(n)}{2}}$.

Exercice n°6 : En utilisant leur décomposition en facteurs premiers, déterminer :

1) La liste des diviseurs de 315.

2) Le nombre de diviseurs de 13 000.

5) Petit théorème de Fermat (hors programme) :

Propriété 7 : Le petit théorème de Fermat.

Soit p un entier premier, et a un entier non multiple de p.

Alors $a^{p-1} \equiv 1 [p]$.

preuve :

Considérons les $p-1$ premiers multiples de a : $a, 2a, 3a, \ldots, (p-1)a$,
puis considérons les restes de ces multiples de a dans la division par p : $r_1, r_2, \ldots, r_{p-1}$.

Montrons que ces restes sont deux à deux distincts :

En effet, s'il existait deux restes identiques, soit r_i et r_j avec $i > j$, alors :

$ia - ja \equiv r_i - r_j [p]$, donc $a(i-j) \equiv 0 [p]$;
donc $(i-j)a$ serait multiple de p, ce qui est impossible,
puisque alors a serait nécessairement multiple de p.

Si ces restes sont tous différents et qu'il y a $p-1$ multiples, c'est que l'on obtient tous les restes non nuls possibles de la division par p, en dehors de 0.

Alors $r_1 \times r_2 \times \ldots \times r_{p-1} = 1 \times 2 \times \ldots \times (p-1) = (p-1)!$

Si l'on cherche le reste du produit de tous ces multiples, on obtient :

$a \times 2a \times \ldots \times (p-1) \equiv (p-1)! [p]$ d'où $(p-1)! a^{p-1} \equiv (p-1)! [p]$

et donc $(p-1)!(a^{p-1}-1) \equiv 0 [p]$.

Comme $(p-1)!$ n'est pas un multiple de p, puisque tous les facteurs sont inférieurs à p, alors $a^{p-1}-1$ est nécessairement un multiple de p.

On a donc $a^{p-1} - 1 \equiv 0 [p]$, ce qui prouve le théorème.

Propriété 8 : Énoncé équivalent.

Pour tout nombre premier p et tout entier a : $a^p \equiv a \left[p \right]$.

preuve :

P7 $\Rightarrow$ P8 :

Si a est multiple de p, alors $a \equiv 0 \left[p \right]$ et donc $a^p \equiv 0 \left[p \right]$.
Si a n'est pas multiple de p, et en multipliant l'égalité $a^{p-1} \equiv 1 \left[p \right]$ par a,
on obtient le résultat.

P8 $\Rightarrow$ P7 :

Comme p divise $a^p - a = a\left(a^{p-1} - 1 \right)$, alors soit p divise a, soit p divise $a^{p-1} - 1$.

Exemple : Prouver que, pour tout entier n, 7 divise $3^{6n} - 1$.

7 est premier et 3 n'est pas un multiple de 7, donc le petit théorème de Fermat nous assure que : $3^6 \equiv 1 \left[7 \right]$
Comme la congruence est compatible avec les puissances, on a : $3^{6n} \equiv 1 \left[7 \right]$,
donc $3^{6n} - 1$ est divisible par 7 pour tout entier n.

Exercice n°7 :

1) Démontrer que pour tout entier n le nombre $n^{13} - n$ est divisible par 26.

2) Démontrer que pour tout entier n le nombre $n^{11} - n$ est divisible par 33.

Propriété 9 : Le petit Théorème de Fermat amélioré.

Soit p et q deux nombres premiers distincts, et $n = pq$.

Pour tout $a \in \mathbb{Z}$ tel que $\text{pgcd}(a, n) = 1$ alors : $a^{(p-1)(q-1)} \equiv 1 \left[n \right]$.

Exemple :

En prenant $p=5$ et $q=7$, on obtient $n=pq=35$ et $(p-1)(q-1)=24$;

$pgcd(a,n)=1$ signifie que p et q ne divisent pas a.

Alors pour tout entier $a=1,2,3,4,6,8,9,11,12,13,14,16,\ldots$ on a $a^{24}\equiv 1[35]$.

preuve :

Tout d'abord, remarquons que $pgcd(a,n)=1 \Leftrightarrow$ p et q ne divisent pas a.
Posons $c=a^{(p-1)(q-1)}$ et calculons c modulo p :

$c=(a^{p-1})^{q-1}\equiv 1^{q-1}[p]\equiv 1[p]$ d'après le petit théorème de Fermat,
puisque p ne divise pas a. De même, on montre que $c\equiv 1[q]$.

Nous allons en déduire que $c\equiv 1(pq)\equiv 1[n]$:

Comme il existe α et β tels que $c=1+\alpha p$ et $c=1+\beta q$, alors $c-1=\alpha p=\beta q$,
ce qui montre que p divise βq.

Mais p et q sont deux nombres premiers distincts, ils sont donc premiers entre eux.
Alors le lemme de Gauss nous assure que p divise β, donc qu'il existe γ tel que $\beta=\gamma p$.

Finalement $c=1+\beta q=1+\gamma pq$, ce qui prouve que $c\equiv 1[pq]$.