

Lien entre les derniers chiffres de la somme et du produit de deux nombres premiers consécutifs (Denise Vella-Chemla, 1/1/2018)

On continue de petites expérimentations de programmation autour des nombres premiers. On a l'idée de calculer pour deux nombres premiers consécutifs p_n et p_{n+1} leur produit et leur somme.

Intriguée par le fait que les derniers chiffres du produit et de la somme semblent "aller de pair", on décide de calculer ce que vaut :

$$10 \times ((p_n \times p_{n+1}) \bmod 10) + ((p_n + p_{n+1}) \bmod 10).$$

On est alors surprise de constater que ce nombre ne semble pouvoir prendre que 13 valeurs différentes qui sont :

$$10, 12, 18, 34, 36, 52, 58, 65, 72, 78, 90, 94, 96.$$

Il faut interpréter ces valeurs ainsi :

- ★ si le produit de deux nombres premiers consécutifs a pour dernier chiffre 1, alors leur somme ne peut avoir pour dernier chiffre que 0, 2 ou 8 ;
- ★ si le produit de deux nombres premiers consécutifs a pour dernier chiffre 3, alors leur somme ne peut avoir pour dernier chiffre que 4 ou 6 ;
- ★ si le produit de deux nombres premiers consécutifs a pour dernier chiffre 5 ou 7, alors leur somme ne peut avoir pour dernier chiffre que 2 ou 8 ;
- ★ si le produit de deux nombres premiers consécutifs a pour dernier chiffre 6, alors leur somme ne peut avoir pour dernier chiffre que 5 ;
- ★ si le produit de deux nombres premiers consécutifs a pour dernier chiffre 9, alors leur somme ne peut avoir pour dernier chiffre que 0, 4 ou 6.

Ce qui surprend le premier de l'an surprend moins le lendemain.

Parmi les 13 valeurs trouvées, on oublie celles commençant ou se terminant par 5 : 65 correspond à $2 \times 3 = 6$ et $2 + 3 = 5$ tandis que 58 est associé aux produit et somme $3 \times 5 = 15, 3 + 5 = 8$ (resp. $52, 5 \times 7 = 35, 5 + 7 = 12$).

On comprend les autres nombres en dressant les tables des derniers chiffres du produit et de la somme pour les nombres premiers autres que 2 ou 5, qui se terminent par 1, 3, 7 ou 9.

×	1	3	7	9
1	1	3	7	9
3	3	9	1	7
7	7	1	9	3
9	9	7	3	1

+	1	3	7	9
1	2	4	8	0
3	4	6	0	2
7	8	0	4	6
9	0	2	6	8

On constate une symétrie pour les derniers chiffres de produit 1 ou 9 : la moitié des produits se terminant par 1 (resp. 9) correspondent à une somme de premiers consécutifs qui se termine par 0 tandis qu'un quart des produits correspondent à une somme se terminant par 2 et un quart des produits correspondent à une somme se terminant par 8 (resp. par 4 ou 6).

On constate une autre symétrie pour les derniers chiffres de produit 3 ou 7 : il semblerait qu'il y ait équilibre (50-50) entre ceux dont la somme se termine par 4 et ceux dont la somme se termine par 6 lorsque le produit se termine par 3 et équilibre également entre ceux dont la somme se termine par 2 et ceux dont la somme se termine par 8 lorsque le produit se termine par 7.

```

1  #include <iostream>
2  #include <stdio.h>
3  #include <cmath>
4
5  int prime(int atester) {
6      bool pastrouve=true;
7      unsigned long k = 2;
8
9      if (atester == 1) return 0;
10     if (atester == 2) return 1;
11     if (atester == 3) return 1;
12     if (atester == 5) return 1;
13     if (atester == 7) return 1;
14     while (pastrouve) {
15         if ((k * k) > atester) return 1;
16         else if ((atester % k) == 0) {
17             return 0 ;
18         }
19         else k++;
20     }
21 }
22
23 int stocke[100] ;
24
25 int main (int argc, char* argv[]) {
26     long int i, p, q, why, a, b, indice ;
27
28     p = 2 ;
29     indice = 1 ;
30     for (i = 1 ; i <= 100 ; ++i)
31         stocke[i] = 0 ;
32     for (i = 3 ; i <= 1000000 ; ++i) {
33         q = i ;
34         if ((prime(p)) && (prime(q))) {
35             std::cout << "\np = " << p ;
36             std::cout << " q = " << q ;
37             a = (p*q) % 10 ;
38             b = (p+q) % 10 ;
39             std::cout << "\npq = " << a << " p+q " << b << "\n" ;
40             why = 10*a + b ;
41             stocke[why] = 1 ;
42             indice = indice+1 ;
43             p = q ;
44         }
45     }
46     for (i = 1 ; i <= 100 ; ++i)
47         if (stocke[i])
48             std::cout << i << " " ;
49 }

```

Fin de l'exécution du programme ci-dessus pour les nombres premiers jusqu'à 10^6

```
1 p = 999883 q = 999907
2 pq = 1 p+q 0
3
4 p = 999907 q = 999917
5 pq = 9 p+q 4
6
7 p = 999917 q = 999931
8 pq = 7 p+q 8
9
10 p = 999931 q = 999953
11 pq = 3 p+q 4
12
13 p = 999953 q = 999959
14 pq = 7 p+q 2
15
16 p = 999959 q = 999961
17 pq = 9 p+q 0
18
19 p = 999961 q = 999979
20 pq = 9 p+q 0
21
22 p = 999979 q = 999983
23 pq = 7 p+q 2
24
25
26
27 10 12 18 34 36 52 58 65 72 78 90 94 96
```