Formule récurrente d'Euler pour le calcul de la somme des diviseurs

Denise Vella-Chemla

4.8.16

1 Etudier la formule de récurrence

Dans l'article *"Découverte d'une loi tout extraordinaire des nombres par rapport à la somme de leurs diviseurs"*, Euler fournit une formule par récurrence qui calcule la somme des diviseurs d'un nombre.

Dans la mesure où la somme des diviseurs permet totalement de distinguer les nombres premiers des nombres composés puisqu'elle vaut $n+1$ pour un nombre n premier alors qu'elle vaut bien davantage pour un nombre composé, on essaie de trouver une nouvelle façon d'envisager cette fonction. On s'interroge notamment sur la manière dont les transformations du carré pourraient intervenir.

Voici la table de la somme des diviseurs des entiers de 1 à 100, présentée en section 4 de la note d'Euler.

$\sigma(1)=1$	$\sigma(21)=32$	$\sigma(41)=42$	$\sigma(61)=62$	$\sigma(81)=121$
$\sigma(2)=3$	$\sigma(22)=36$	$\sigma(42)=96$	$\sigma(62)=96$	$\sigma(82)=126$
$\sigma(3)=4$	$\sigma(23)=24$	$\sigma(43)=44$	$\sigma(63)=104$	$\sigma(83)=84$
$\sigma(4)=7$	$\sigma(24)=60$	$\sigma(44)=84$	$\sigma(64)=127$	$\sigma(84)=224$
$\sigma(5)=6$	$\sigma(25)=31$	$\sigma(45)=78$	$\sigma(65)=84$	$\sigma(85)=108$
$\sigma(6)=12$	$\sigma(26)=42$	$\sigma(46)=72$	$\sigma(66)=144$	$\sigma(86)=132$
$\sigma(7)=8$	$\sigma(27)=40$	$\sigma(47)=48$	$\sigma(67)=68$	$\sigma(87)=120$
$\sigma(8)=15$	$\sigma(28)=56$	$\sigma(48)=124$	$\sigma(68)=126$	$\sigma(88)=180$
$\sigma(9)=13$	$\sigma(29)=30$	$\sigma(49)=57$	$\sigma(69)=96$	$\sigma(89)=90$
$\sigma(10)=18$	$\sigma(30)=72$	$\sigma(50)=93$	$\sigma(70)=144$	$\sigma(90)=234$
$\sigma(11)=12$	$\sigma(31)=32$	$\sigma(51)=72$	$\sigma(71)=72$	$\sigma(91)=112$
$\sigma(12)=28$	$\sigma(32)=63$	$\sigma(52)=98$	$\sigma(72)=195$	$\sigma(92)=168$
$\sigma(13)=14$	$\sigma(33)=48$	$\sigma(53)=54$	$\sigma(73)=74$	$\sigma(93)=128$
$\sigma(14)=24$	$\sigma(34)=54$	$\sigma(54)=120$	$\sigma(74)=114$	$\sigma(94)=144$
$\sigma(15)=24$	$\sigma(35)=48$	$\sigma(55)=72$	$\sigma(75)=124$	$\sigma(95)=120$
$\sigma(16)=31$	$\sigma(36)=91$	$\sigma(56)=120$	$\sigma(76)=140$	$\sigma(96)=252$
$\sigma(17)=18$	$\sigma(37)=38$	$\sigma(57)=80$	$\sigma(77)=96$	$\sigma(97)=98$
$\sigma(18)=39$	$\sigma(38)=60$	$\sigma(58)=90$	$\sigma(78)=168$	$\sigma(98)=171$
$\sigma(19)=20$	$\sigma(39)=56$	$\sigma(59)=60$	$\sigma(79)=80$	$\sigma(99)=156$
$\sigma(20)=42$	$\sigma(40)=90$	$\sigma(60)=168$	$\sigma(80)=186$	$\sigma(100)=217$

Et voilà la formule de récurrence qu'Euler a trouvée.

$$\begin{cases} \sigma(n)=\sigma(n-1)+\sigma(n-2)-\sigma(n-5)-\sigma(n-7)+\sigma(n-12)+\sigma(n-15) \\ \qquad\quad -\sigma(n-22)-\sigma(n-26)+\sigma(n-35)+\sigma(n-40-\sigma(n-51)-\sigma(n-57) \\ \qquad\quad +\sigma(n-70)+\sigma(n-77)-\sigma(n-92)-\sigma(n-100)+etc. \\ \sigma(0)=\sigma(1)=1 \\ \sigma(n)=0 \text{ si n} < 0. \end{cases}$$

Plusieurs éléments entrent en ligne de compte :

- les nombres pentagonaux n_k qui sont à soustraire à n pour savoir quels $\sigma(n - n_k)$ utiliser pour appliquer la formule de récurrence.

  Euler fournit l'aide suivante :

  la progression des nombres 1, 2, 5, 7, 12, 15, etc. qu'il faut successivement retrancher du nombre proposé n, deviendra évidente, en prenant leurs différences :

N.	1,	2,	5,	7,	12,	15,	22,	26,	35,	40,	51,	57,	70,	77,	92,	100, *etc.*
Diff.	1,	3,	2,	5,	3,	7,	4,	9,	5,	11,	6,	13,	7,	15,	8,	*etc.*

  Car alternativement, on aura tous les nombres naturels 1, 2, 3, 4, 5, 6, etc. *et les nombres impairs* 3, 5, 7, 9, 11, etc.*, d'où l'on pourra continuer la suite de ces nombres aussi loin qu'on voudra.*

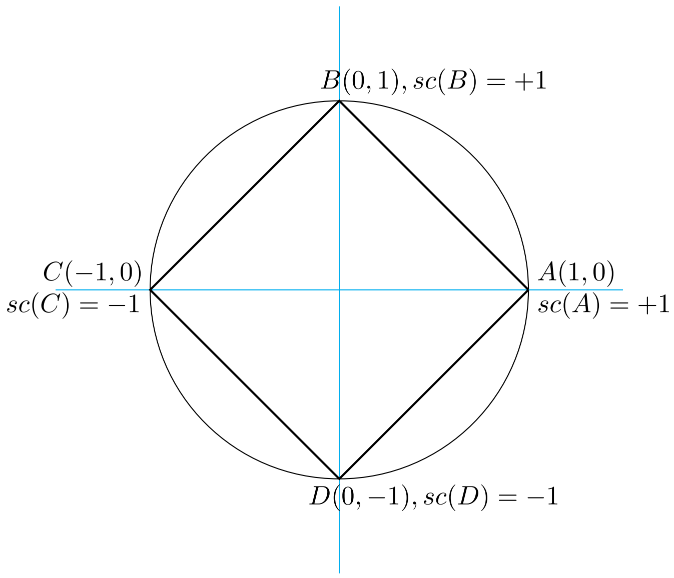
- le fait d'alterner deux signes "+" et deux signes "$-$" pour agréger les différents $\sigma(n - n_k)$;

- le fait qu'il faille remplacer $\sigma(0)$ par n et $\sigma(n - n_k)$ par 0 dans le cas où $n - n_k$ est négatif.

Pour le calcul des nombres pentagonaux, on utilise la série définie par :

$$\begin{cases} s_1 & = 1, \\ s_n & = s_{n-1} + n/2 \qquad \forall n > 1 \text{ pair} \\ & = s_{n-1} + n \qquad \forall n > 1 \text{ impair} \end{cases}$$

Pour alterner deux signes "+" suivis de deux signes "$-$" cycliquement, on va utiliser une rotation de $\pi/2$ (antihoraire) sur le dessin suivant :



et utiliser la somme des coordonnées qui vaudra alternativement $+1, +1, -1, -1, +1, +1, -1, \ldots$ On nécessite la série de nombres complexes z_n définie par :

$$\begin{cases} z_1 & = 1, \\ z_n & = i z_{n-1} \end{cases}$$

(z_1 correspond au point A et les points sont parcourus cycliquement selon l'ordre $A, B, C, D, A, B, C, D, etc.$).

On a aussi besoin de la fonction somme des coordonnées sc définie sur les complexes par $sc(a + ib) = a + b$.

La récurrence pour la somme des diviseurs s'écrit alors :

$$\begin{cases} \sigma(n) & = 0 & \forall n < 0 \\ \sigma(0) & = n & \forall n \ (hum\ !), \\ \sigma(1) & = 1, & \\ \sigma(n) & = \sum_{k=1}^{n} sc(z_k)\sigma(n - s_k) & \forall n > 1 \end{cases}$$

2 Programmer

Est fourni page suivante le programme proposé.

```
#include <iostream>
#include <complex>
#include <cmath>
#include <stdio.h>

#define M_PI 3.14159265358979323846

typedef std::complex<double> dcomplex ;
const dcomplex di = dcomplex(0.0,1.0) ;

int main (int argc, char* argv[])
{
  int n, tempo, sommedivtempo, k, nmax ;
  int s[350] ;
  int sigma[350] ;
  int sc[350] ;
  dcomplex z[350] ;

  nmax = 320 ;
  s[1] = 1 ;
  for (n = 2 ; n <= nmax ; ++n) {
    if ((n % 2) == 0) s[n] = s[n-1]+n/2 ;
    else s[n] = s[n-1]+n ;
  }
  z[1] = dcomplex(1.0,0.0) ;
  for (n = 2 ; n <= nmax ; ++n) z[n] = di * z[n-1] ;
  for (n = 1 ; n <= nmax ; ++n) sc[n] = real(z[n]) + imag(z[n]) ;
  sigma[1] = 1 ;
  for (n = 2 ; n <= nmax ; ++n) {
    tempo = n ;
    sommedivtempo = 0 ;
    k = 1 ;
    while (tempo-s[k] >= 0) {
      if (tempo-s[k] == 0) sommedivtempo = sommedivtempo + sc[k] * n ;
      else sommedivtempo = sommedivtempo + sc[k] * sigma[tempo - s[k]] ;
      k=k+1 ;
    }
    sigma[n] = sommedivtempo ;
    std::cout << n << " --> " << sigma[n] << "\n" ;
  }
}
```

On a l'intuition de la manière dont peuvent intervenir ces signes qui alternent en se plaçant sur un corps seulement, modulo 5 par exemple. $13 \; mod \; 5 = 3$ car 13 est compris entre $10 = 2 \times 5$ et $15 = 3 \times 5$ et $13 - 10 = 3$. Si on prend deux produits kx et $(k+1)x$ avec $k \geqslant 3$ et qu'on leur ôte 13, ces différences seront de même signe (positif). Si on prend deux produits kx et $(k+1)x$ avec $k < 2$ et qu'on leur ôte 13, les différences seront de même signe (négatif). Le seul cas où les différences kx et $(k+1)x$ sont de signe opposé est le cas où $k = 3 = 13 \; div \; 5$ (div désignant ici la division entière). Quand on fait le produit de l'infinité des corps premiers $\mathbb{Z}/p\mathbb{Z}$, ces changements de signes interfèrent les uns avec les autres et la récurrence de la somme des diviseurs d'Euler agrège tous ces changements par un simple parcours cyclique de $\mathbb{Z}/4\mathbb{Z}$ et changement de signe une fois sur 2 seulement.

Pour rappel même si cela n'a pas été utilisé ici, $\sigma(n)$ est une fonction multiplicative.

Pour rappel également, l'élévation à la puissance (suite géométrique) ou bien le fait d'additionner toujours une même raison (suite arithmétique) font "cycler" les restes dans un corps premier (la même suite de tous les restes permutés dans un certain ordre revient cycliquement indéfiniment).

On rappelle enfin que $x^{a+ib} = x^a cos(b \; ln \; x) + ix^a sin(b \; ln \; x)$.

Alors la motivation est grande de transformer cette formule de calcul de la somme des diviseurs pour obtenir une formule qui fournirait directement par de simples opérations booléennes le caractère de primalité d'un entier donné (on imagine que les caractères de primalité à utiliser pourraient être les mêmes que ceux utilisés ici pour calculer la somme des diviseurs).

Il faut revenir sur le nombre de termes qui intervient dans chaque étape de la récursion : on n'est pas satisfait du critère d'arrêt (noté d'un "hum !" ci-dessus) qui fait s'arrêter la formule récurrente quand la fonction σ a comme argument 0 ou bien un nombre entier négatif.

On étudie plus avant le nombre de termes et on obtient que la somme des termes intervenant dans le calcul de $\sigma(n)$ contient :

- 1×1 terme pour 1 ;
- 3×2 termes pour 2, 3 et 4 ;
- 2×3 termes pour 5 et 6 ;
- 5×4 termes pour 7, 8, 9, 10 et 11 ;
- 3×5 termes pour 12, 13 et 14 ;
- 7×6 termes pour 15, 16, 17, 18, 19, 20 et 21 ;
- 4×7 termes pour 22, 23, 24 et 25 ;
- 9×8 termes pour 26, 27, 28, 29, 30, 31, 32, 33 et 34 ;
- 5×9 termes pour 35, 36, 37, 38 et 39 ;
- 11×10 termes pour 40, 41, 42, 43, 44, 45, 46, 47, 48, 49 et 50 ;
- etc.

Cela peut être résumé par :

- il y a $\dfrac{p+1}{2}$ nombres entiers successifs dont la somme récurrente contient p termes lorsque p est impair ;
- il y a $p+1$ nombres entiers successifs dont la somme récurrente contient p termes lorsque p est pair.

On voit bien à nouveau les deux termes du produit faire alterner les pairs et les impairs successifs s'agissant du premier, ou bien parcourir la suite simple des entiers successifs en ce qui concerne le deuxième.

Il est étonnant que le nombre de termes de la somme récurrente $\sigma(n)$ à calculer soit fixé étant donné n. Il se calcule ainsi :

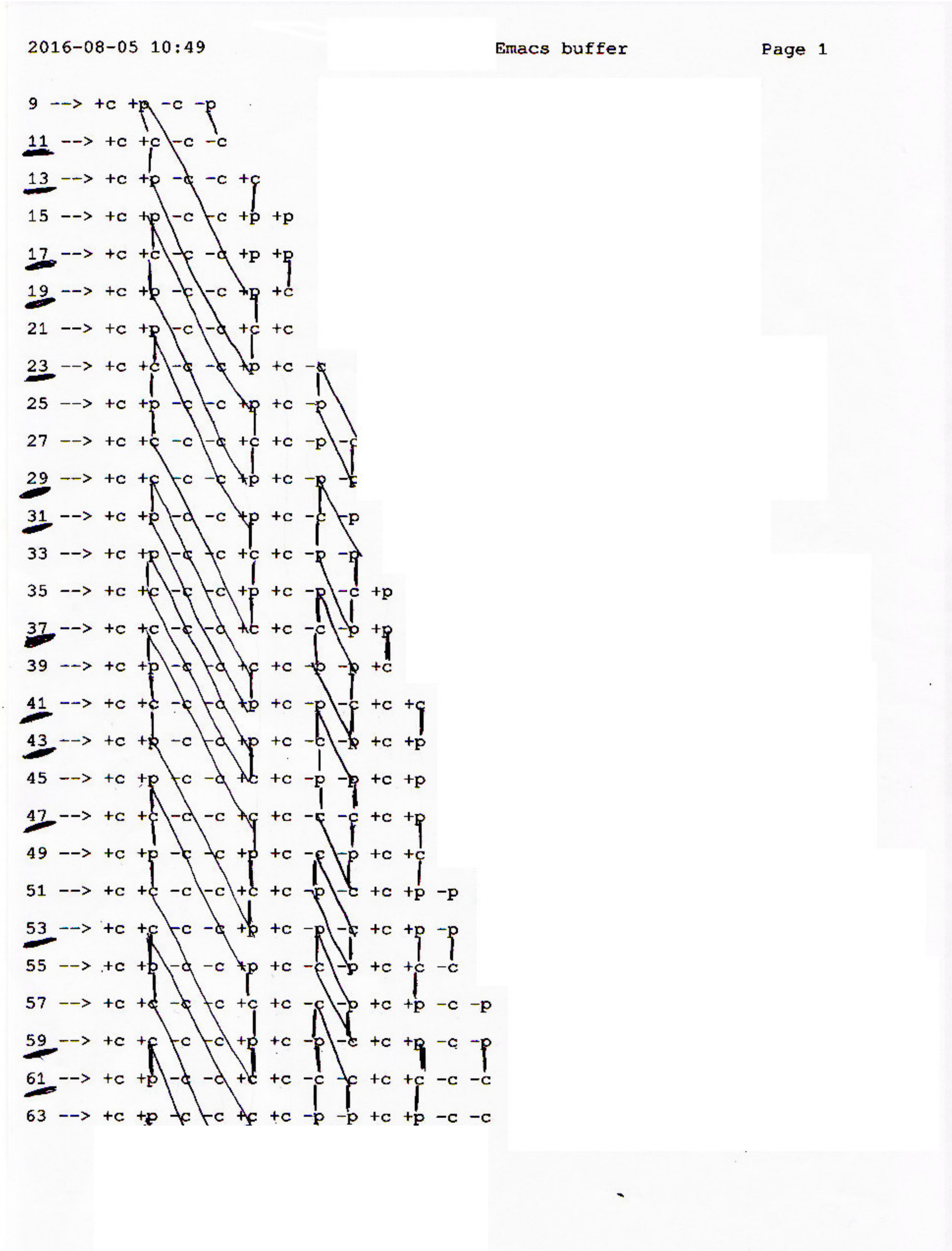
- trouver le plus grand entier inf_1 inférieur à n de la forme $f(x) = \dfrac{3x^2-x}{2}$ ainsi que le plus petit entier sup_1 de cette forme qui lui soit supérieur ;
- trouver le plus grand entier inf_2 inférieur à n de la forme $g(x) = \dfrac{3x^2+x}{2}$ ainsi que le plus petit entier sup_2 de cette forme qui lui soit supérieur ;
- si $f(inf_1) \leq n < f(inf_2)$ alors il y aura $2inf_1 - 1$ termes intervenant dans la formule ;
- si $g(inf_2) \leq n < (sup_1)$ alors il y aura $2inf_1$ termes intervenant dans la formule.

Ainsi, pour 30, compris entre 22 et 35 selon la fonction f et entre 26 et 40 selon la fonction g, étant supérieur à 26, il y aura 8 termes intervenant dans la formule récurrente de calcul de $\sigma(30)$.

3 Détourner le programme à la recherche de relations invariantes

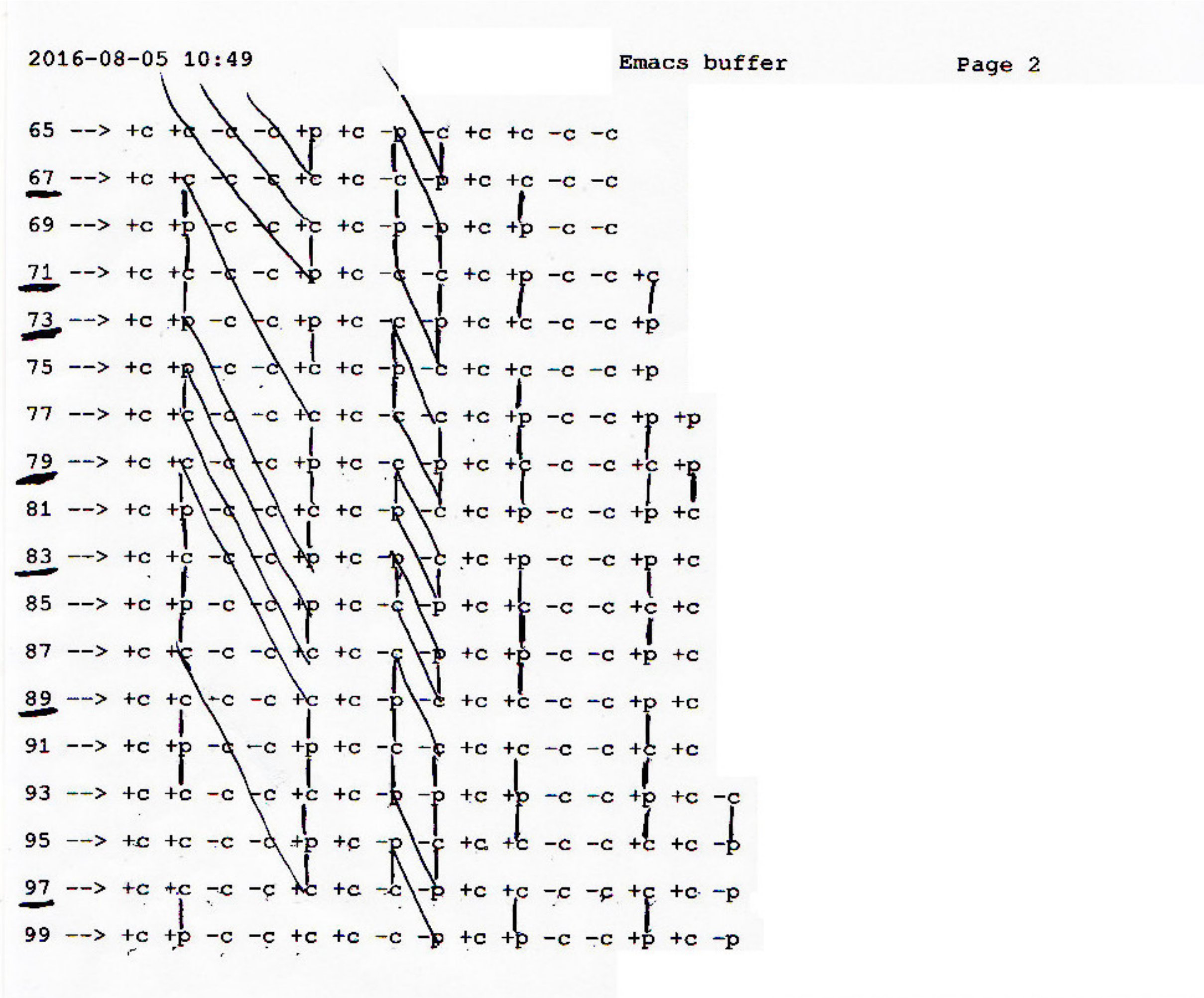
On modifie le programme de manière à ce qu'il écrive p ou c selon que les $\sigma(n-n_i)$ intervenant dans la récurrence sont premiers ou composés. On conserve les signes alternés une fois sur 2 "+,+,-,-,+,+,..." et on effectue des comptages : combien obtient-on de $+c$, de $+p$, de $-c$ et de $-p$? On compte, ce faisant, et comme on l'a déjà fait à une autre occasion, des nombres d'assertions logiques.

On scanne le fichier résultant.



```
2016-08-05 10:49                     Emacs buffer        Page 1

9 --> +c +p -c -p
11 --> +c +c -c -c
13 --> +c +p -c -c +c
15 --> +c +p -c -c +p +p
17 --> +c +c -c -c +p +p
19 --> +c +p -c -c +p +c
21 --> +c +p -c -c +c +c
23 --> +c +c -c -c +p +c -c
25 --> +c +p -c -c +p +c -p
27 --> +c +c -c -c +c +c -p -c
29 --> +c +c -c -c +p +c -p -c
31 --> +c +p -c -c +p +c -c -p
33 --> +c +p -c -c +c +c -p -p
35 --> +c +c -c -c +p +c -p -c +p
37 --> +c +c -c -c +c +c -c -p +p
39 --> +c +p -c -c +c +c -p -p +c
41 --> +c +c -c -c +p +c -p -c +c +c
43 --> +c +p -c -c +p +c -c -p +c +p
45 --> +c +p -c -c +c +c -p -p +c +p
47 --> +c +c -c -c +c +c -c -c +c +p
49 --> +c +p -c -c +p +c -c -p +c +c
51 --> +c +c -c -c +c +c -p -c +c +p -p
53 --> +c +c -c -c +p +c -p -c +c +p -p
55 --> +c +p -c -c +p +c -c -p +c +c -c
57 --> +c +c -c -c +c +c -c -p +c +p -c -p
59 --> +c +c -c -c +p +c -p -c +c +p -c -p
61 --> +c +p -c -c +c +c -c -c +c +c -c -c
63 --> +c +p -c -c +c +c -p -p +c +p -c -c
```



```
2016-08-05 10:49                         Emacs buffer            Page 2

65 --> +c +c -c -c +p +c -p -c +c +c -c -c
67 --> +c +c -c -c +c +c -c -p +c +c -c -c
69 --> +c +p -c -c +c +c -p -p +c +p -c -c
71 --> +c +c -c -c +p +c -c -c +c +p -c -c +c
73 --> +c +p -c -c +p +c -c -p +c +c -c -c +p
75 --> +c +p -c -c +c +c -p -c +c +c -c -c +p
77 --> +c +c -c -c +c +c -c -c +c +p -c -c +p +p
79 --> +c +c -c -c +p +c -c -p +c +c -c -c +c +p
81 --> +c +p -c -c +c +c -p -c +c +p -c -c +p +c
83 --> +c +c -c -c +p +c -p -c +c +p -c -c +p +c
85 --> +c +p -c -c +p +c -c -p +c +c -c -c +c +c
87 --> +c +c -c -c +c +c -c -p +c +p -c -c +p +c
89 --> +c +c -c -c +c +c -p -c +c +c -c -c +p +c
91 --> +c +p -c -c +p +c -c -c +c +c -c -c +c +c
93 --> +c +c -c -c +c +c -p -p +c +p -c -c +p +c -c
95 --> +c +c -c -c +p +c -p -c +c +c -c -c +c +c -p
97 --> +c +c -c -c +c +c -c -p +c +c -c -c +c +c -p
99 --> +c +p -c -c +c +c -c -p +c +p -c -c +p +c -p
```

On repère des parallélogrammes contenant les mêmes lettres, dûs au fait que lorsque un $n - n_i$ intervenant dans l'un des calculs est égal au $n' - n'_i$ intervenant dans un autre, le caractère de primalité est le même puisqu'il code l'information concernant un même nombre dans les deux cas. On ne trouve pas de critère général qui permettrait de distinguer les nombres premiers des nombres composés, pourtant les informations fournies ici devraient suffire puisque la somme des diviseurs, calculée selon une récurrence similaire, contient le caractère de primalité d'un nombre quant à elle. Peut-être qu'en ne gardant que le caractère de primalité plutôt que la somme des diviseurs de chaque nombre, on a perdu de l'information.