Parité (Denise Vella-Chemla, 13.5.2017)

On cherche des formules les plus simples possible pour calculer les valeurs des variables du tableau :

n	$\pi(n)$	$\Delta(n)$	$S(n)$	$C(n)$	$R(n)$		n	$\pi(n)$	$\Delta(n)$	$S(n)$	$C(n)$	$R(n)$		n	$\pi(n)$	$\Delta(n)$	$S(n)$	$C(n)$	$R(n)$
1	0	0	0	0	0		34	11	2	60	58	37		67	19	0	161	161	114
2	1	0	0	0	0		35	11	2	62	60	38		68	19	4	165	161	114
3	2	0	0	0	0		36	11	7	69	62	39		69	19	2	167	165	117
4	2	1	0	-1	0		37	12	0	69	69	45		70	19	6	173	167	118
5	3	0	1	1	0		38	12	2	71	69	45		71	20	0	173	173	123
6	3	2	3	1	0		39	12	2	73	71	46		72	20	10	183	173	123
7	4	0	3	3	1		40	12	6	79	73	47		73	21	0	183	183	132
8	4	2	5	3	1		41	13	0	79	79	52		74	21	2	185	183	132
9	4	1	6	5	2		42	13	6	85	79	52		75	21	4	189	185	133
10	4	2	8	6	2		43	14	0	85	85	57		76	21	4	193	189	136
11	5	0	8	8	3		44	14	4	89	85	57		77	21	2	195	193	139
12	5	4	12	8	3		45	14	4	93	89	60		78	21	6	201	195	140
13	6	0	12	12	6		46	14	2	95	93	63		79	22	0	201	201	145
14	6	2	14	12	6		47	15	0	95	95	64		80	22	8	209	201	145
15	6	2	16	14	7		48	15	8	103	95	64		81	22	3	212	209	152
16	6	3	19	16	18		49	15	1	104	103	71		82	22	2	214	212	154
17	7	0	19	19	10		50	15	4	108	104	71		83	23	0	214	214	155
18	7	4	23	19	10		51	15	2	110	108	74		84	23	10	224	214	155
19	8	0	23	23	13		52	15	4	114	110	75		85	23	2	226	224	164
20	8	4	27	23	13		53	16	0	114	114	78		86	23	2	228	226	165
21	8	2	29	27	16		54	16	6	120	114	78		87	23	2	230	228	166
22	8	2	31	29	17		55	16	2	122	120	83		88	23	6	236	230	167
23	9	0	31	31	18		56	16	6	128	122	84		89	24	0	236	236	172
24	9	6	37	31	18		57	16	2	130	128	89		90	24	10	246	236	172
25	9	1	38	37	23		58	16	2	132	130	90		91	24	2	248	246	181
26	9	2	40	38	23		59	17	0	132	132	91		92	24	4	252	248	182
27	9	2	42	40	24		60	17	10	142	132	91		93	24	2	254	252	185
28	9	4	46	42	25		61	18	0	142	142	100		94	24	2	258	254	186
29	10	0	46	46	28		62	18	2	144	142	100		95	24	2	258	256	187
30	10	6	52	46	28		63	18	4	148	144	101		96	24	10	268	258	188
31	11	0	52	52	33		64	18	5	153	148	104		97	25	0	268	268	197
32	11	4	56	52	33		65	18	2	155	153	108		98	25	4	272	268	197
33	11	2	58	56	36		66	18	6	161	155	109		99	25	4	276	272	200
														100	25	7	283	276	203

On a :

- $S(n) = \displaystyle\sum_{i=2}^{n-3} \left(\left\lfloor \frac{n}{i} \right\rfloor - 1 \right)$;
- $\Delta(n) = \#\{d < n \text{ tel que } d \mid n\}$; dans l'article A032741[1] de l'OEIS (Open Encyclopedia of Integer Sequences), il est indiqué que $a(n) = \Delta(n) + 1$ (pour $n \geqslant 7$) est le nombre de facteurs du nième polynôme de Fibonacci ou que $\Delta(n+1)$ est le nombre de facteurs du polynôme du nième degré $x^n + x^{n-1} + x^{n-2} + \ldots$; si $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k}$ alors $\Delta(n) = -2 + \displaystyle\prod_{i=1}^{k} (\alpha_i + 1)$;
- $C(n)$ est le nombre de produits d'entiers ij strictement inférieurs à n, pour i et j variant de 2 à $n-2$;
- on impose un ordre total lexicographique sur les produits de deux entiers : ab est avant $a'b'$ selon cet ordre si et seulement si $(a < a')$ *ou* $((a = a')$ *et* $(b < b'))$.

  $R(n)$ (pour nombre de redondances) est le nombre de produits d'entiers $i'j'$ strictement inférieurs à n, pour i' et j' variant de 2 à $n-2$ tels qu'il existe un produit de même valeur ij avec (i, j) qui est antérieur à (i', j') selon l'ordre lexicographique : on ne garde qu'un représentant par classe de produits de deux entiers de même valeur ;

[1] séquence à démarrer à 2.

- $C(n) \ = \ S(n) - \Delta(n)$;
- $\pi(n) \ = \ n - S(n) + \Delta(n) + R(n) - 2 = n - C(n) + R(n) - 2$;
- $C(n+1) = S(n)$ pour $n \geqslant 5$;
- $S(n) + \Delta(n+1) = S(n+1)$ pour $n \geqslant 5$;
- $S(n) = \sum\limits_{k=1}^{n} \Delta(k)$ et $C(n) = \sum\limits_{k=1}^{n-1} \Delta(k)$ pour $n \geqslant 5$.

Exemple :

$C(10) = 6$ car 6 est le nombre de produits de la liste ci-dessous strictement inférieurs à 10 (on les a notés en rouge).

2×2	2×3	2×4	2×5	2×6	2×7	2×8
3×2	3×3	3×4	3×5	3×6	3×7	3×8
4×2	4×3	4×4	4×5	4×6	4×7	4×8
5×2	5×3	5×4	5×5	5×6	5×7	5×8
6×2	6×3	6×4	6×5	6×6	6×7	6×8
7×2	7×3	7×4	7×5	7×6	7×7	7×8
8×2	8×3	8×4	8×5	8×6	8×7	8×8

$R(10) = 2$ car $3 \times 2 = 2 \times 3$ et $4 \times 2 = 2 \times 4$: on dénombre 2 produits redondants. L'ordre lexicographique dont il a été question plus haut est ici l'ordre de lecture habituel, de gauche à droite puis de bas en haut.

On constate et il faudrait le démontrer que $S(n)$ et $S(n-1)$ n'ont pas même parité lorsque n est un carré d'entier (une puissance paire d'entier).

On constate et il faudrait le démontrer que $R(n)$ et $R(n+1)$ ont même parité lorsque n est un nombre premier ou un carré d'entier (une puissance paire d'entier).

On constate et il faudrait le démontrer que $S(n) = C(n)$ lorsque n est un nombre premier. C'est une condition surprenante :

$$n \ \textit{premier}$$
$$\Longleftrightarrow$$
$$\sum_{i=2}^{n-3} \left(\left\lfloor \frac{n}{i} \right\rfloor - 1 \right) = \# \left\{ xy \ \ tels \ \ que \ \ (xy < n) \ \wedge \ (2 \leqslant x \leqslant n-2) \ \wedge \ (2 \leqslant y \leqslant n-2) \right\}$$

Cette condition d'égalité $S(n) = C(n)$ fait vraiment souhaiter voir les nombres premiers comme points fixes d'une fonction qui associerait $S(n)$ à $C(n) = S(n-1)$ (i.e. ferait passer de $S(n-1)$ à $S(n)$).

On a utilisé le programme fourni page suivante.

```
#include <iostream>
#include <stdio.h>
#include <math.h>
#include <vector>
#include <bitset>

int prime(int atester)
{ bool pastrouve=true; unsigned long k = 2;

  if (atester == 1) return 0;
  if (atester == 2) return 1;
  if (atester == 3) return 1;
  if (atester == 5) return 1;
  if (atester == 7) return 1;
  while (pastrouve) {
    if ((k * k) > atester) return 1;
    else if ((atester % k) == 0) return 0 ; else k++;
  }
}

int main(int argc, char* argv[]) {
  std::vector<bool> dejatrouve(n) ;
  int n, i, j, pix, compteprod, compteproddessous ;
  int gardenbredondances, nbredondances, gardesomme, somme, nbdiv ;
  int schangeparite, redondchangeparite, nbimpairs ;

  for (n = 1 ; n <= 1000 ; ++n)
    {
      pix = 0 ;
      compteproddessous = 0 ;
      gardenbredondances = nbredondances ;
      nbredondances = 0 ;
      gardesomme = somme ;
      somme = 0 ;
      nbdiv = 0 ;
      for (i = 2 ; i < n ; ++i) dejatrouve[i] = false ;
      for (i = 2 ; i < n ; ++i) if (prime(i)) pix=pix+1 ;
      for (i = 2 ; i <= n/2 ; ++i) if (n % i == 0) nbdiv = nbdiv+1 ;
      for (i = 2 ; i < n-2 ; ++i) somme = somme+((n/i)-1) ;
      for (i = 2 ; i <= n-2 ; ++i)
         for (j = 2 ; j <= n-2 ; ++j) {
           if (i*j < n) {
             compteproddessous = compteproddessous+1 ;
             if (dejatrouve[i*j] == false) dejatrouve[i*j] = true ;
             else nbredondances = nbredondances+1 ;
           }
         }
      std::cout << "\nn -> " << n << "\n" ;
      std::cout << "pi(x) = " << pix << "\n" ;
      std::cout << "nbdiv " << nbdiv << "\n" ;
      std::cout << "gardesomme " << gardesomme << "\n" ;
      std::cout << "somme " << somme << "\n" ;
      if (((somme % 2) == 0) != ((gardesomme % 2) == 0))
        std::cout << "Somme change de parité. \n" ;
      std::cout << "compteproddessous " << compteproddessous << "\n" ;
      std::cout << "gardenbredondances " << gardenbredondances << "\n" ;
      std::cout << "nbredondances " << nbredondances << "\n" ;
      if (((gardenbredondances % 2)==0) == ((nbredondances % 2)==0))
        std::cout << "nbredondances ne change pas de parité pour " << n-1 << "\n" ;
    }
}
```