Redondire (Denise Vella-Chemla, 15.4.2017)

On voudrait présenter ici une ébauche d'élaboration d'un comptage exact du nombre de nombres premiers inférieurs à un nombre donné. On a procédé de maière expérimentale, en programmant certains calculs. Comme dans quelques notes récentes au sujet d'hyperboles, on s'intéresse à l'écriture des nombres comme produits de deux entiers supérieurs ou égaux à 2. On a utilisé le programme suivant :

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

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[]) {
  const int n=45000 ;
  int i, j, pix, compteprod, compteproddessous, nbredondances, somme, nbdivpropres ;
  std::vector<bool> dejatrouve(n) ;

  pix = 0 ; compteproddessous = 0 ; nbredondances = 0 ;
  somme = 0 ; nbdivpropres = 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) nbdivpropres = nbdivpropres+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 ;
        //std::cout << i << "*" << j << "=" << i*j << "\n" ;
        if (dejatrouve[i*j] == false) dejatrouve[i*j] = true ;
        else nbredondances = nbredondances+1 ;
      }
    }
  std::cout << "pi(x) = " << pix << "\n" ;
  std::cout << "somme " << somme << "\n" ;
  std::cout << "compteproddessous " << compteproddessous << "\n" ;
  std::cout << "nbdivpropres " << nbdivpropres << "\n" ;
  std::cout << "nbredondances " << nbredondances << "\n" ;
  std::cout << n-compteproddessous+nbredondances << "\n" ;
}
```

La variable *pix* compte le nombre de nombres premiers strictement inférieurs à n. La variable *compteproddessous* compte le nombre de produits de la forme xy avec x et y compris entre 2 et $n-2$ qui sont strictement inférieurs à n. La variable *nbdivpropres* compte le nombre de diviseurs propres de n (i.e. différents de 1 et n). La variable *somme* est égale à $\sum_{i=2}^{n-3} \left(\left\lfloor \frac{n}{i} \right\rfloor - 1 \right)$.

Voici les résultats des calculs effectués par le programme pour quelques nombres :

n	$\pi(x)$	$somme$	$compteproddessous$	$nbdivpropres$	$nbredondances$	$n - compteproddessous + nbredondances$
100	25	283	276	7	203	27
1000	168	5070	5056	14	4226	170
10000	1229	73669	73646	23	64877	1231
45000	4675	399133	399075	58	358752	4677

On a systématiquement :

- $nbdivpropres + compteproddessous = somme$
- ainsi que $pix = n - compteproddessous + nbredondances - 2$.

C'est normal : les nombres premiers sont ceux qui ne peuvent pas s'écrire sous la forme d'un produit de deux nombres supérieurs ou égaux à 2 quels qu'ils soient.

Notre problème reste entier, on n'a fait que le déplacer :

- pourrait-on trouver une formule exacte pour le nombre de produits inférieurs strictement à n ? (comptés par la variable *compteproddessous* du programme et dont on voit qu'il est égal à $-nbdivpropres(n) + \sum_{i=2}^{n-3} \left(\left\lfloor \frac{n}{i} \right\rfloor - 1 \right)$)
- comment compter exactement les redondances, i.e. le nombre d'égalités de la forme $a.b = c.d$?