

*Programme préféré : les nombres premiers annulent une somme de cosinus
(Denise Vella-Chemla, 21.5.2016)*

Le programme en C++ ci-dessous, à compléter par un test de nullité (si ce n'est qu'en informatique, la nullité d'un flottant est testée à ε près), distingue les nombres premiers des nombres composés par calcul d'une somme de cosinus.

```
1 #include <iostream>
2 #include <cmath>
3 #include <stdio.h>
4 #define _USE_MATH_DEFINES
5
6 int main (int argc, char* argv[]) {
7     float res ;
8     int n, o, b ;
9
10    for (n = 1 ; n <= 100 ; ++n) {
11        printf("%d -> ", n) ;
12        res = 0.0 ;
13        for (b = 2 ; b <= n-1 ; ++b) {
14            for (o = 1 ; o <= b ; ++o) {
15                res=res+cos(2.0*M_PI*(float)n*(float)o/(float)b) ;
16            }
17        }
18        printf("%f\n", res) ;
19    }
20 }
```

Voici son résultat pour les entiers impairs jusqu'à 99. Le calcul renvoie 0 pour les nombres premiers et renvoie la somme des diviseurs propres de x pour x composé.

```
1 1 -> 0.000000
2 3 -> 0.000000
3 5 -> 0.000000
4 7 -> 0.000000
5 9 -> 3.000000
6 11 -> 0.000000
7 13 -> 0.000000
8 15 -> 8.000000
9 17 -> -0.000000
10 19 -> 0.000000
11 21 -> 10.000004
12 23 -> 0.000000
13 25 -> 5.000001
14 27 -> 12.000004
15 29 -> 0.000000
16 31 -> 0.000000
17 33 -> 14.000007
18 35 -> 12.000004
19 37 -> 0.000000
20 39 -> 16.000004
21 41 -> -0.000000
22 43 -> 0.000000
23 45 -> 31.999992
24 47 -> 0.000000
25 49 -> 7.000001
26 51 -> 20.000011
27 53 -> -0.000001
28 55 -> 16.000008
29 57 -> 21.999996
30 59 -> 0.000001
31 61 -> 0.000001
32 63 -> 39.999992
33 65 -> 17.999973
34 67 -> 0.000001
35 69 -> 26.000006
36 71 -> 0.000000
37 73 -> -0.000002
38 75 -> 47.999992
39 77 -> 18.000011
40 79 -> 0.000002
41 81 -> 38.999973
42 83 -> 0.000001
43 85 -> 21.999985
44 87 -> 32.000008
45 89 -> 0.000002
46 91 -> 19.999964
47 93 -> 34.000004
48 95 -> 23.999977
49 97 -> -0.000002
50 99 -> 55.999832
```