

```

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

using namespace std ;

#define M_PI 3.14159265358979323846
typedef std::complex<double> dcomplex ;
const dcomplex di = dcomplex(0.0,1.0) ;

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[])
{
    dcomplex somme ;
    int k, detantentant ;

    somme = 0.0 ;
    detantentant = 0 ;
    for (k = 1 ; k < 1000 ; ++k)
        if (prime(k)) {
            detantentant = detantentant+1 ;
            somme = somme + exp(2.0*di*M_PI / (double) k) ;
            std::cout << k << " --> " << somme << "\n" ;
        }
}

```