```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Source: complex.cpp
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     1  // $Id: complex.cpp,v 1.12 2013-08-09 15:51:26-07 - - $
     2
     3  #include <complex>
     4  #include <iomanip>
     5  #include <iostream>
     6  using namespace std;
     7
     8  typedef complex<double> dcomplex;
     9
    10  #define D(X,V) cout << "const dcomplex " #X " = " #V ";" << endl; \
    11                 const dcomplex X = V; \
    12                 cout << #X << " = " << X << endl;
    13
    14  int main() {
    15     cout << setprecision(15);
    16     D(C_N1,-1);
    17     D(C_I,sqrt(C_N1));
    18     D(C_PI,M_PI);
    19     D(C_I_PI,C_I * M_PI);
    20     D(C_EXP,exp (C_I_PI));
    21     D(E_I_PI_P1,C_EXP + 1.0);
    22     return 0;
    23  }
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Output: ./complex 2>&1
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
     1  const dcomplex C_N1 = -1;
     2  C_N1 = (-1,0)
     3  const dcomplex C_I = sqrt(C_N1);
     4  C_I = (0,1)
     5  const dcomplex C_PI = M_PI;
     6  C_PI = (3.14159265358979,0)
     7  const dcomplex C_I_PI = C_I * M_PI;
     8  C_I_PI = (0,3.14159265358979)
     9  const dcomplex C_EXP = exp (C_I_PI);
    10  C_EXP = (-1,1.22460635382238e-16)
    11  const dcomplex E_I_PI_P1 = C_EXP + 1.0;
    12  E_I_PI_P1 = (0,1.22460635382238e-16)
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
pstatus: 0x0000 EXIT STATUS = 0
```