



```

// file: j0_test.cpp
//
// C++ Program to demonstrate use of the J0 Bessel function from
// the gsl numerical library.
//
// Programmer: Dick Furnstahl furnstahl.i@osu.edu
//
// Revision history:
// 12/26/03 original C++ version, modified from C version
// 01/03/05 switched to <cmath>
//
// Notes:
// * Example taken from the GNU Scientific Library Reference Manual
// Edition 1.1, for GSL Version 1.1 9 January 2002
// URL: gsl/ref/gsl-ref_23.html#SEC364
// * Compile and link with:
// g++ -Wall -o J0_test J0_test.cpp -lgsl -lgslibcblas
// * gsl routines have built-in
// extern "C" {
//     <header stuff>
// }
// so they can be called from C++ programs without modification
// * The answer should be J0(5) = -1.775967713143382920e-01
//
//*****
// include files
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cmath>
using namespace std;

#include <gsl/gsl_sf_bessel.h> // gsl Bessel special function header file

int
main ()
{
    double x = 5.0; // just a random test value

    double y = gsl_sf_bessel_J0 (x); // see the GSL manual for details

    cout << "J0(" << x << ")=" << endl;
    cout << setprecision(18) << setw(20) << y << endl;

    return 0;
}

```

```

// file: precision.cpp
//
// This program determines machine precision e
// (the smallest e for which 1 + e does not equal 1)
// Be careful when interpreting the output!
//
// Programmer: Dick Furnstahl furnstahl.i@osu.edu
//
// Revision history:
// 02-Jan-2004 original version, for 780.20 Computational Physics
// 02-Jan-2005 changed name from limit to precision for clarity,
// added test and made one a const
// 01-Jan-2007 added file output so usable on Windows
//
// Notes:
// * compile with: "g++ -o precision precision.cpp"
// * adapted from: "Projects in Computational Physics" by Landau and Paez
//   copyrighted by John Wiley and Sons, New York
//   code copyrighted by RH Landau
// * comment: very crude program which produces lots of output
//
//*****
// include files
#include <iostream> // basic input/output functions
#include <iomanip> // manipulators like setprecision
using namespace std; // we need this when .h is omitted
//*****
const int max_iterations = 600;

int
main ()
{
    float eps = 1.0; // starting value
    const float one = 1.0;

    ofstream my_out ("precision.out"); // open a "stream" to precision.out
    for (int i = 0; i < max_iterations; i++)
    {
        eps /= 1.1; // divide by a fixed factor
        float test = one + eps; // is "test" different from 1.0?
        // output 12 digits in "test" to file
        my_out << setprecision (12) << test << " " << eps << endl;
    }

    my_out.close(); // close the output stream
    return 0; // successful completion
}

```