Program to test when programs compiled with g++ return nan's and inf's

Programmer: Dick Furnstahl furnstahl.1@osu.edu

Revision history:
01/24/04 original version, translated from nan_test.c
02/01/04 added log(-1.)
01/20/06 added examples of combining nan's and inf's

Notes:
* "nan" stands for "not a number"
* "inf" stands for infinity
* Error removed from make_nan_test to allow it to proceed with
  warning (which may not occur anyway)
* If we explicitly put 0. in a denominator, we would get a compiler
  warning. But defining denominator=0. and then using it gets
  no warning! (So you can't rely on the compiler for this.)

To do:

#include files
#include <iostream>    // note that .h is omitted
#include <iomanip>     // note that .h is omitted
#include <string>
using namespace std;
#include <cmath>

************************** main program ***************************
int main (void)
{
  cout << "Fun and games with nan's and inf's: " << endl;
  double numerator = 1.
  double denominator = 0.;
  cout << "1./0. ==> " << numerator/denominator << endl;
  cout << "-1./0. ==> " << -numerator/denominator << endl;
  cout << " log(0.)==> " << log(0.) << endl;
  cout << " sqrt(-1.)== " << sqrt(-1.) << endl;
  cout << " log(-1)== " << log(-1.) << endl;

  // Now let's try some predictions
  cout << endl << "Now try predicting. In each case, predict nan or inf."
  string answer;    // define a C++ string to hold the answer
  cout << " log(exp(exp(10.)))==> ";
  cin >> answer;
  cout << "You answered " << answer << ". The correct answer is: "
  << log(exp(exp(10.))) << endl;
  cout << " arccos(2)== ";
  cin >> answer;
  cout << "You answered " << answer << ". The correct answer is: "
  << acos(2.) << endl;
  // Now let's try combining
  double my_inf = numerator/denominator;
  double my_nan = sqrt(-1.); return 0;
}