SIN FUNCTION

// Author: Jerry Olds

// ID: 1001533643

// Class: CSE-3310-002

// Assignment 1

#define BOOST\_TEST\_DYN\_LINK

#define BOOST\_TEST\_MODULE MyTest

#include <iostream>

#include <ctime>

#include <boost/test/unit\_test.hpp>

#include <boost/random.hpp>

// This function will calculate the sin of the given value

double sin\_func(float X, int \*errorflag)

{

// Validate user input

if(X < 0.0 || X > 2 \* M\_PI)

{

\*errorflag = 1;

return -1;

}

// Calculate sin using Taylor series

return X - (pow(X,3)/6) + (pow(X,5)/120) - (pow(X,7)/5040) +

(pow(X,9)/362880) - (pow(X,11)/39916800);

}

TEST CASES

// Author: Jerry Olds

// ID: 1001533643

// Class: CSE-3310-002

// Assignment 1

// My rationale that the function works is that when you send 0 to the function it returns 0,

// when you send PI/2 it returns 1, when you send PI it returns 0, and when you send

// 3PI/2 it returns -1. So the function returns the correct crest, trough, and equlibrium position

// of sin curve

#include "jdo3643\_sin.cpp"

// (psuedo) Black box testing. Will generate a random floating point value between -3 and 10

// and will return the sin of the random value if it is in the range of 0 and 2 \* PI

BOOST\_AUTO\_TEST\_CASE(TestCase1)

{

// Error flag for if random value is not in the valid range

int errorflag = 0;

// Create random number

boost::random::mt19937 num(std::time(0));

boost::random::uniform\_real\_distribution<> dist{-3, 10};

float input = dist(num);

std::cout << "\nGenerating random real number between -3 and 10..." << std::endl;

std::cout << "Random number generated is " << input << "." << std::endl;

// Send random value to sin function

double result = sin\_func(input, &errorflag);

// Output result if input is outside of valid range

if(errorflag == 1)

{

if(input < 0)

std::cout << input << " is less than 0. Cannot compute sin(" << input << ")" << std::endl << std::endl;

else

std::cout << input << " is greater than " << 2 \* M\_PI << ". Cannot compute sin(" << input << ")" << std::endl << std::endl;

}

// Output result of value in valid range

else

{

std::cout << "sin(" << input << ") using Taylor series of 6 terms = " << result << std::endl;

std::cout << "sin(" << input << ") = " << sin(input) << std::endl;

BOOST\_CHECK\_CLOSE(result, sin(input), 1.0f);

}

}

// White box testing. Will generate a random floating point value between 0 and 2 \* PI

// and will return the sin of the random value

BOOST\_AUTO\_TEST\_CASE(TestCase2)

{

// Error flag for if random value is not in the valid range

int errorflag = 0;

// Create random number

boost::random::mt19937 num(std::time(0));

boost::random::uniform\_real\_distribution<> dist{0, 2 \* M\_PI};

float input = dist(num);

std::cout << "\n\nGenerating random real number between 0 and " << 2 \* M\_PI << "..." << std::endl;

std::cout << "Random number generated is " << input << "." << std::endl;

// Send random value to sin function

double result = sin\_func(input, &errorflag);

// Output result

std::cout << "sin(" << input << ") using Taylor series of 6 terms = " << result << std::endl;

std::cout << "sin(" << input << ") = " << sin(input) << std::endl;

}