#include <cmath>

#include <string>

#include <iostream>

using namespace std;

bool OddEven(int i) {//checks to see if the input is even or odd

double i\_out = i / 2;

if (i\_out == (int)i\_out)

return false;

return true;

}

double functionMaker(string function, double x) { //Takes the string of a function and

int length = function.length();

for (int i = 0; i < function.length(); i++) {

char\* char\_x = (char\*)(&x);

if (function[i] == 'x')

function[i] = \*char\_x;

}

return pow(x, 0.5);

}

double SimpsonsRule(int a, int b, int n) {

double delta\_x = (((double)b - (double)a) / (double)n);

double sum\_out = 0, x = a;

for (int i = 1; i < n; i++) {

double function\_Sum = functionMaker("Ignore", x);

if (i != 1 && i != n) {

bool isOdd = OddEven(i);

if (isOdd == true)

function\_Sum = function\_Sum \* 4;

else

function\_Sum = function\_Sum \* 2;

}

sum\_out += function\_Sum;

x = x + delta\_x;

}

return ((delta\_x / 3) \* sum\_out);

}

void errorHandling() {//executes the incorrect error process when the user inputs an incorrect value

cin.clear(); //clear bad input flag

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); //discard input

cout << "Invalid input; please re-enter." << endl;

}

int main() {

int min\_bound, max\_bound, no\_intervals;

/\*string function;

cout << "Enter function to be used in Simpson's Rule: " << endl;

cin >> function;\*/

while (cout << "Enter minimum bound of the function: " << endl && !(cin >> min\_bound))errorHandling();

while (cout << "Enter maximum bound of the function: " << endl && !(cin >> max\_bound)) errorHandling();

while (cout << "Finally enter the number of intervals: " << endl && !(cin >> no\_intervals)) errorHandling();

cout << "The result of integrationg is " << SimpsonsRule(min\_bound, max\_bound, no\_intervals) << endl;

return 0;

}