Aishwarya Ravishankar

Yamin S Yee

Jacob Goryance

Prime Number

Input:

#include <iostream>

#include <cmath>

using namespace std;

/\* This main function prompts the user for the number of prime numbers to find. Then goes into a while loop using an int loop counter which is the primeSuspect and checks to see if its a prime number

\*/

int main() {

int primeTotal;

cout << "please enter the amount of prime numbers you would like to see:" << endl;

cin >> primeTotal; /\* prompt user for amount of prime numbers \*\*/

double primeSuspect = 2.0; /\* initialize primeSuspect = 2, becuase 1 cannot be possible\*/

int primeFound =0;

bool isPrime(int); /\* set boolean \*/

while (true) { /\* if prime is true primeFound counter goes increments by 1 \*\*/

int intPrimeSuspect = (int) primeSuspect;

if (isPrime(intPrimeSuspect)) {

cout << "the prime root is: " << primeSuspect << " and its square root is: " << sqrt (primeSuspect) << endl; /\* find square root \*\*/

primeFound ++;

}

primeSuspect+=1.0;

if ( primeFound == primeTotal) {

break; /\* stop and completed successfully \*/

}

}

cout << "completed Succesfully" << endl;

return 0;

}

/\* returns if number is prime or not \*\*/

bool isPrime(int suspect){

   int highest = sqrt(suspect);

for ( int i = 2; i < suspect ; i ++){

 if ((suspect) %i == 0){

  return false;

}

}

return true;

}

/\* This double precision square root function takes a double precison input, and produces a double precision output.

\*/

double sqrt(double num)

{

double lower\_bound=0;

double upper\_bound=num;

double temp=0; /\* ek edited this line \*/

int nCount = 50; /\* declares all the varaibles in terms of double\*/

/\* When the while statement is true it compares and sets lower\_bound and upper\_bound or else returns temp\*/

while(nCount != 0)

{

temp=(lower\_bound+upper\_bound)/2;

if(temp\*temp==num)

{

return temp;

}

else if(temp\*temp > num)

{

upper\_bound = temp;

}

else

{

lower\_bound = temp;

}

nCount--;

}

return temp;

}

/\* This square root function takes a long integer input, and produces a long integer output

\*/

long int sqrt (long input) {

if (input < 0) {

cout << "you cannot take the root of a negative number";

return -1; /\* if its less than -1 it returns -1\*/

}

else { /\* if the statement is false it declares the varaibles and it takes the square root\*/

int root =(input/2);

int difference;

int mid;

while (true){

difference =(input-(root\*root));

difference = abs(difference);

if ( difference < (pow(10,-6))){

return root;

}

else {

root = ((input/root)+root)/2;

}

}

}

}

Output:

