/\*

Name: David Giacobbi

Class: CPSC 122, Section 01

Date Submitted: January 18, 2021

Assignment: Project 1

Description: This program generates a list of the first prime numbers and displays them in a certain amount of columns, determined by the user's command line argument.

\*/

#include <iostream>

**using** **namespace** std;

//Function prototype declaration

**bool** isPrime(**int**);

**void** display(**int**, **int**);

**void** error(**int**);

**int** main(**int** argc, **char**\* argv[])

{

//Declare int variables for number of primes and columns to be displayed in console

**int** totalPrimes = atoi(argv[1]);

**int** cols = atoi(argv[2]);

//Error checking for totalPrimes and cols

**if**(cols < 1)

error(0);

**if**(totalPrimes < 1)

error(1);

**if**(totalPrimes < cols)

error(2);

//Display request prime numbers in amount of columns as directed by user

display(totalPrimes, cols);

}

/\*

Description: Determines whether input integer is prime

Input: integer whose primality is to be judged

Returns: true if num is prime, false otherwise

\*/

**bool** isPrime(**int** num)

{

//For loop implemented to check each number less than the number being checked to see if it is divisible by any other than itself

**for**(**int** i = 2; i < num; i++)

{

//Condition would invalidate a number's ability to be prime

**if**(num % i == 0)

{

**return** **false**;

**break**;

}

}

**return** **true**;

}

/\*

Description: Loops over all necessary candidate primes, invoking isPrime on each, displaying in column fashion those that are prime

Input: integer, totalPrimes, indicating the number of primes to display; integer cols, indicating over how many columns the primes are to be displayed

Return: N/A

\*/

**void** display(**int** totalPrimes, **int** cols)

{

//Count assures that the amount of primes requested are printed (regulates while loop); checkPrime is the number being checked to see if passes the prime test

**int** count = 0;

**int** checkPrime = 2;

**while**(count < totalPrimes)

{

**if**(isPrime(checkPrime))

{

cout << checkPrime << '\t';

**if**(count % cols == cols - 1)

cout << endl;

count++;

}

checkPrime++;

}

}

/\*

Description: Provides error message for different invalid inputs

Input: integer, key, determines which error message should print based on what error has occurred

Return: N/A

\*/

**void** error(**int** key)

{

**if**(key == 0)

{

cout << "The number of columns must be one or greater." << endl;

exit(EXIT\_FAILURE);

}

**if**(key == 1)

{

cout << "The number of prime numbers must be one or greater." << endl;

exit(EXIT\_FAILURE);

}

**if**(key == 2)

{

cout << "There must be more prime numbers than columns requested for display." << endl;

exit(EXIT\_FAILURE);

}

}