

Programming Project 02

This assignment is worth 10 points and must be **completed and turned in before 11:59 on Monday, September 18th, 2017.**

Assignment Overview

This assignment will exercise your ability to utilize control statements (`while`, `for`, `if`) for C++

Background

We are going to look at highly composite numbers

https://en.wikipedia.org/wiki/Highly_composite_number. A highly composite number is calculated in the following way. For the positive integer n

- We calculate the count of divisors, numbers that divide evenly into n without remainder.
- We also calculate the count of divisors for every number from 1 to $n-1$
- We say that n is highly composite if it has a count of divisors greater than the count of divisors of any of the integers 1 to $n-1$

The Wikipedia page gives a list of highly composite numbers of different orders. The order of the numbers lists the next element in the sequence of integers that increases its count of divisors. The column $d(n)$ gives the count of divisors.

For example, 12 is a highly composite number. Its divisors are: 1, 2, 3, 4, 6, 12. It has the order 5 (5th in the series from 1 that increase its divisor count). No number 1-11 has the same or more divisors than 12.

Another example, 20 is not highly composite. It has as its divisors 1, 2, 4, 5, 10, 20. The first number from 1 to 20 that has 6 divisors is the number 12, as we just saw.

Project Description / Specification

Input:

- Input is a single, positive integer that is 1 or greater.

Output for each test case will be:

- If the input is 0 or less, then the word "Error" is printed and processing ends.
- If no error, then you must determine if the input is highly composite
 - If it is highly composite, print "True", a space, the input number, a space, and the count of divisors. For 12, the output would be `True 12 6` on a single line.
 - If it is not highly composite, print "False", a space, the input number, a space, the first number in the sequence 1 to input that has at least the same number of divisors, a space, and the count of the smaller number's divisors. For 20, the output would be `False 20 12 6` on a single line. 20 has 6 divisors (1, 2, 4, 5, 10, 20) but 12 was the first number in the sequence 1 to 20 with 6 divisors.

Requirements

1. As mentioned, any input number from the test case is not an integer ≥ 1 prints "Error". Though the requirements also mention that it should be an integer, we are not yet capable of testing for that.

Deliverables

proj02.cpp -- your source code solution (remember to include your section, the date, project number and comments).

1. Please be sure to use the specified file name, i.e. “**proj02.cpp**”
2. Electronically submit a copy of the file to Mimir
 - 1) For Mimir, note that there are 4 test cases for 2 points each, a total of 8 points. 2 more points will be assessed after you turn the project in by the TAs for the quality of your code.

Notes

1. Don't get all weirded out by the testing of Mimir. Figure out each part of the program and test it yourself. Can you, for example, determine if the input number is ≥ 1 ? Write that code and test it yourself. Mimir is the final test, but trying to do it all at once is a mistake.
2. The cycle should be edit-compile-test. Do it in the small, work your way up, and only after you have tested all the parts, then test it against Mimir.
3. You can do all the work in x2go, in Mimir (before you submit it for testing), on your own laptop, doesn't matter. Get your rhythm going, find out your workflow and start to see if you have a good approach. These are easy problems, they get harder so work on your approach now while you have a chance.
4. Finally, stay on top of this class! The worst part about getting behind here is that it is very difficult to catch up. Look at the examples, watch the videos, look at the slides. If you miss something (a project, a pre-lab, etc.) brush it off and keep going. There is no makeup but there is a lot of material and you have many opportunities. Don't let one late element get you down, just keep going!