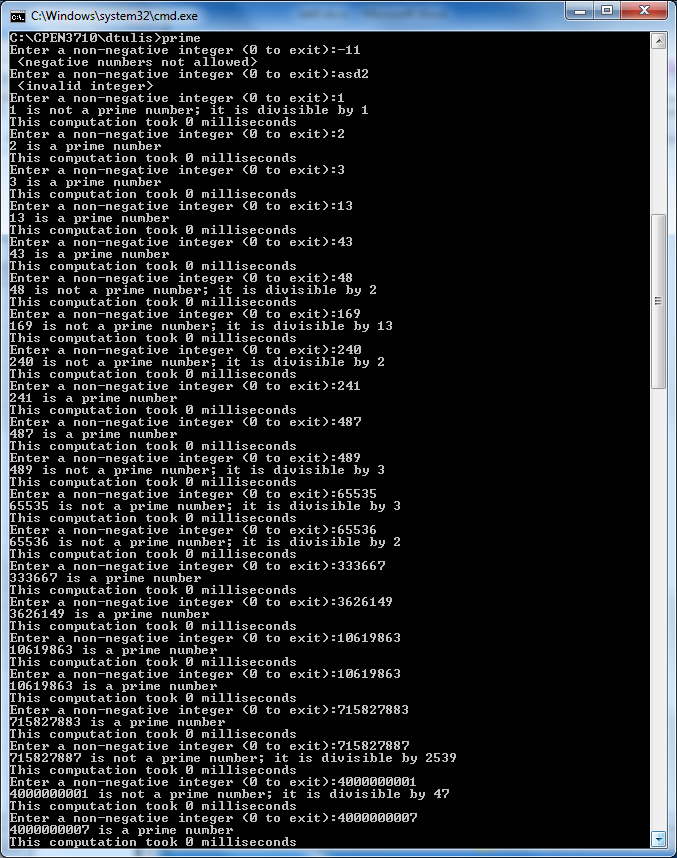
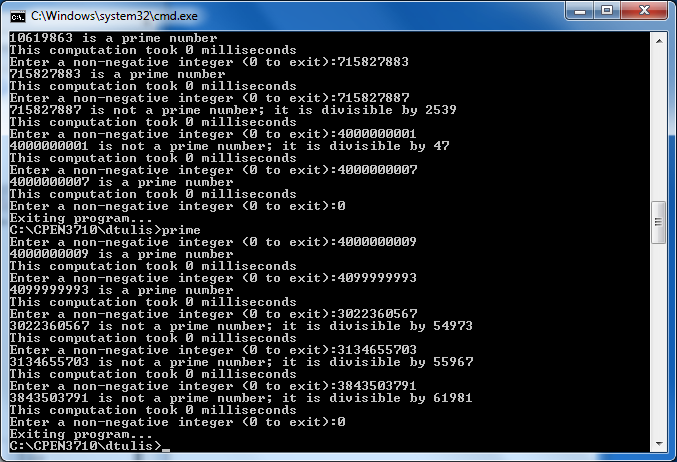
David Tulis bqb428

CPEN3710 lab 9



Some addition testing of numbers with only 2 large prime factors.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Run number | Number to test | # of digits | Is prime? | Test time |
| 1 | 1 | 1 | no | 0 |
| 2 | 2 | 1 | yes | 0 |
| 3 | 3 | 1 | yes | 0 |
| 4 | 13 | 2 | yes | 0 |
| 5 | 43 | 2 | yes | 0 |
| 6 | 48 | 2 | no | 0 |
| 7 | 169 | 3 | no | 0 |
| 8 | 240 | 3 | no | 0 |
| 9 | 241 | 3 | yes | 0 |
| 10 | 487 | 3 | yes | 0 |
| 11 | 489 | 3 | no | 0 |
| 12 | 65535 | 5 | no | 0 |
| 13 | 65536 | 5 | no | 0 |
| 14 | 333667 | 6 | yes | 0 |
| 15 | 3626149 | 7 | yes | 0 |
| 16 | 10619863 | 8 | yes | 0 |
| 17 | 715827883 | 9 | yes | 0 |
| 18 | 715827887 | 9 | no | 0 |
| 19 | 4000000001 | 10 | no | 0 |
| 20 | 4000000007 | 10 | yes | 0 |

This data does not give us any information on the how long it takes to test for primality. Our program works with 32 bit unsigned integers (232 – 1, or 4,294,967,295). We tested almost to the boundaries of our data type, and the time to test was too small to be measured in milliseconds.

There are two things we could do to get better data:

1. If we tested unsigned 64 bit integers (264 – 1, or 18,446,744,073,709,551,615), we might have been able to see some worthwhile data.
2. If instead of measuring milliseconds (1 thousandth of a second), we could have measured in microseconds (1 millionth of a second). Unfortunately, Irvine does not provide a way to test this, so we are stuck with measuring in milliseconds.

Obviously, the larger the number gets, the longer it will take to test, but this test cannot be done with the limits listed above.