

Due Thursday, 23 Sep 1999, at the beginning of class.

1 Reading

Finish reading Chapter 2 of the textbook. If you desire, start reading Chapter 3.

2 Problems

When submitting a program, please submit a printed copy of the program's code prepended with comments briefly describing its input and output. Please indent your code to make it readable. See Section 2.5 of the textbook for style hints. If you desire, you can also submit examples demonstrating your program's correctness.

1. Texas population data, ordered according to county, appears in <http://www.cs.trinity.edu/~joldham/1320/hw/hw3/tx-county-data>. There is one line per county, and each line has three positive integers:
 - (a) the county's 1990 population,
 - (b) the county's 1998 July population estimate, and
 - (c) the county's 1999 January population estimate.

Write a program that prints

- (a) the state population in 1990,
 - (b) the estimated state population in 1998 July,
 - (c) the estimated state population in 1999 January,
 - (d) the largest county population in 1990,
 - (e) the largest estimated county population in 1998 July, and
 - (f) the largest estimated county population in 1999 January.
2. In Norton Juster's children's story *The Phantom Tollbooth*, the Mathemagician gives Milo the following problem to solve:

$$4 + 9 - 2 * 16 + 1 / 3 * 6 - 67 + 8 * 2 - 3 + 26 - 1 / 34 + 3 / 7 + 2 - 5.$$

According to Milo's calculations, which are corroborated by the Mathemagician, this expression "all works out to zero." If you do the calculation, however, the expression comes out to zero only if you start at the beginning and apply all the operators in strict left-to-right order. What would the answer be if the Mathemagician's expression were evaluated using C++'s precedence rules?

In your answer, show the order the operations are performed. An easy way to do this is to copy the entire expression and then replace the performed operation by its value. Copy the resulting expression to the next line and continue. (Using a text editor to copy the expression may be useful.)

3. Write a computer program asking the user for a number of seconds and converting them to more normal notation, e.g., years, days, hours, minutes, and seconds. For example, 31622401 seconds equals 1 year, 1 day, 0 hours, 0 minutes, and 1 second. For your calculation, ignore leap years and leap seconds.
4. Write a computer program that computes the greatest common divisor (gcd) of two nonnegative integers. The greatest common divisor is the largest positive integer that evenly divides both integer. For example, 4 is the greatest common divisor of 8 and 12.

When we reduce fractions to lowest terms, we implicitly compute gcds. For example, the fraction $24/36$ equals $2/3$ because both the numerator and denominator can be evenly divided by 12. 2 and 3 have no common divisors so the fraction is in lowest terms and 12 is the greatest common divisor.

Here is Euclid's algorithm to compute gcd of two integers a and b , assuming $a \geq b$: If $b = 0$, then the gcd is a . Otherwise, the gcd equals the gcd of b and the remainder of a divided by b .

- (a) By hand, use the algorithm to compute the gcd of 16 and 28. Submit your work showing how you computed the gcd.
- (b) Write a C++ program to compute the gcd of any two nonnegative integers.