CSE 232, Lab Exercise 8

Partner

Choose a partner in the lab to work with on this exercise. Two people should work at one computer. Occasionally switch who is typing.

The Problem

You remember the Collatz sequence, don't ya? Here is a little reminder in case you didn't. http://en.wikipedia.org/wiki/Collatz_conjecture One of the problems is that, by doing the calculation over from the beginning for every number, you waste a lot of time redoing work that already has been done. Look at the first 6 sequences:

1	1
2	2, 1
3	3, 10, 5, 16, 8, 4, 2, 1
4	4, 2, 1
5	5, 16, 8, 4, 2, 1
6	6, 3, 10, 5, 16, 8, 4, 2, 1

Subsequences, whose value we already know, are recalculated. For example, the entire sequence of 3 is already known, but we recalculate the entire sequence when starting from 6.

Memoization

Memoization (note the spelling, no "r") is a simple optimization technique used in algorithms where repeat calculations occur (http://en.wikipedia.org/wiki/Memoization). The idea is simple. We *remember* a sequence when we first calculate it. Subsequently, for any value we calculate we check first to see if we already know the remaining sequence. If so, we stop the calculation and just copy the known sequence in.

For this lab, we will use a map to memorize Collatz sequences. Pay attention to this technique, it comes up in many guises and is very useful!

Programming Task

Make a new project directory called `lab08`. Copy `lab08_functions.h` to your directory. You need to write both `lab08_functions.cpp` and `lab08_main.cpp`. You will turn in `lab08_functions.cpp` to Mimir for testing.

Data structure, `map` of `long` to `vector<long>`

The data structure you will use for the lab is a map that has:

- as a **key**, a `long` representing the first number in the Collatz sequence

- as a **value**, a `vector<long>` representing the resulting Collatz sequence if you started at the key number

The declaration of such a map would look like:

- `map<long, vector<long>> collatz_map;`

Maps store their elements as an STL `pair`. When you iterate through a map you iterate through a sequence of `pair`. The `pair` type for the map above would be:

- `pair<long, vector<long>> collatz_element;`

Remember that, given such a `pair`, `collatz_element.first` is the key (the `long`) and `collatz_element.second` is the value (the `vector<long>` sequence).

For convenience, we have provided a using statement in the header

```
using Collatz = pair<long, vector<long> >;
```

Functions
The various functions, their inputs and their outputs are provided in `lab08_functions.h` Read the descriptions and use them to solve the problem.

Extra

If you get done early and wish to, trying printing out the three longest sequences in the range you just examined. This is shown on the example output.

- Hint! It is very simple if you use algorithms like `sort`, `transform` and write a compare function for the sort

Assignment Notes

1. Use `find` or `count`, not `[ ]` to see if a key is in the map
    a. remember that, by using the `[ ]` to search a map, you ***always*** find the key because it will insert that key if it is not there. To see if something is in a map you have to use `find`.
    b. `find` returns either an iterator to the found element or `collatz_map.end()` if it did not find it.
    c. The iterator that `find` returns is a pointer to a pair. Remember the fun syntax I showed you: `(*itr).first` is the same as `itr->first`
2. To append something to the end of a vector:
    a. you could write a loop
    b. you could use `copy` and a `back_inserter`