COMP 40 Lab: A compiler sprint

Introduction

This week, we want to see how you are doing on compiler errors. We will give you a program that contains errors; your goal is to get it to compile. *Speed counts.*

Preparing for the lab

*Please complete this lab working individually. If you have come with a partner, you should each follow the instructions below which will result in your making individual, submissions.*

Before starting, you may want to set up your editor to track compiler error messages so that you can compile and immediately jump to the erroneous source line within your editor. Some hints on doing this can be found at the COMP 40 reference page on [Compiling and finding errors with as few keystrokes as possible](#). Of course, there are many other hints and tricks that you can find on the Web.

When you are ready to begin, get the code by running

```
   git clone /comp/40/git/compiler-sprint
```

You should get three files: `diagram.c`, `compile`, and `nyt`.

Debugging your program and submitting

Your mission is simple: *run the compile script repeatedly until it succeeds.* The compile script itself should submit your source code and your times.

The compiled program, `diagram`, finds potential solutions to a diagramless crossword puzzle. To see it in action, you can run, e.g.,

```
   ./diagram 19 nyt
```

A note for Emacs users

You are welcome to use Vim, Emacs or any other professional grade editor you like. (If you are using Kate, it may be too late to switch for this lab, but you should definitely figure out soon whether you prefer Vim,
Emacs, or some similar high-quality tool for professionals. Almost surely you want a tool that allows you to run compile scripts within the editor, and makes it easy to jump to the source lines with the errors.)

One way to do this in Emacs is:

- Use Alt-X compile (I.e. type Alt-X and then compile) to bring up a compile prompt
- Backspace over the default make -l command to replace it with sh compile (you’ll only have to do this the first time)
- Press enter to run the compile script
- If there are errors, use CTRL-` (that is: hold down CTRL and press x. Then release both and press the backquote key. Each time you do that Emacs will take you to the next erroneous source line.
- to get back to one window, click in the source window and type CTRL-X 1

This should make your debugging much more efficient, however: Emacs isn’t entirely prepared for the fact that when your program succeeds the compile script will prompt you for input. Your best bet is to get out of Emacs and rerun the compile script from a command prompt. That will do the submission but may warn you that it’s your second submission. Don’t worry about it. Just go ahead and submit.