

# Homework 5

*Due: 9 April 2012*

## Problem 5.1

Write a series of test files. At least one of them should have an `int main()` as its only top-level definition (no global variables or other functions), and at least one of them should have multiple functions defined and no `main`. (There can be more than two if you like.) All should be both syntactically and semantically valid within the Project 4 subset.

Among all of them there should be examples of every different kind of AST node, and examples that make use of each different parser function (e.g. not just one binary operator, but an addition, and also a multiplication, and so on). Basically, running the parser and (eventually) the semantic analysis on these files should test every code path except the errors.

## Problem 5.2

Run `gcc -S` on each of the files you wrote for the previous problem. Print out the resulting assembly file alongside the original C file, and annotate the assembly by indicating which fragments of assembly correspond to which fragments of C. You may find it helpful to tweak something in the `.c` and see what changes in the `.s`. Be as detailed and/or verbose as you can; this will help you later on when you plan a roadmap for your code generation.

Tip: If you run

```
enscript -U2 -E -o test1.ps test1.c test1.s
```

on torvalds it will create a file `test1.ps` with side-by-side pretty-printed code from the two other files. You can then use `scp` or `PSFTP` or the equivalent to get the file to a local machine to print it.