

### CSE 756, Project 3: simpleC Pretty Printer with ROSE

You need to build a pretty printer for a subset of C. This will be done by traversing the Sage AST in ROSE, extracting the necessary information from the AST, and constructing a string that contains the pretty printing of the program. This string will be printed by `main` (already provided at the web page; do **not** change it). The result should be a valid C program that is equivalent to the input program. The project is due by **April 30 (Saturday)**, 11:59 pm.

#### Input Language

Assume that the input is a C program that satisfies all restrictions described in Project 2, with the following exception. For Project 3, we will extend the language with *if-statements* and *if-else-statements*, as described in Section 6.8.4.1 of the ANSI C document. Of course, ROSE can handle general C programs (as well as C++ and Fortran program), but we will only consider this particular subset of C. In addition, there are several simplifying assumptions on loops, as described under “Notes on AST notes” at the web page; *please read these notes very carefully*.

#### Output

The result should be a single string value that is returned back to the `main` function provided on the web page (do **not** change this `main`; it will be used by the grader). This string should be a valid C program that can be compiled (by `gcc` on `stdsun`) and executed. The behavior of this program should be exactly the same as the behavior of the original program.

#### Details

- Your submission must compile and run on `stdlogin`, using the ROSE installation in `/class/cse756/...`
- During the analysis of the AST, do **not** print directly to `cout`. The only printing to `cout` must be done by the provided `main`.
- Do **not** use the ROSE methods `unparseToString` and `unparseToCompleteString`.
- Your submission must work correctly on the two test programs used for Project 2.
- The web page contains a sample file `pretty.cpp`, with some useful examples of AST processing, as well as some useful notes on the different kinds of AST nodes. *Please read carefully the code in this file.*

#### Project Submission

On or before 11:59 pm on the due date, you should submit a single file `pretty.cpp` containing all of your code; it should work with the provided `main`. Submit your project using “`submit c756aa lab3 .`” or “`submit c756aa lab3 pretty.cpp`” on `stdsun`.

If the timestamp on your electronic submission is **12:00 am on the next day or later**, you will receive 10% reduction per day, for up to three days. If your submission is later than 3 days after the deadline, it will not be accepted and you will receive zero points for this project. If you resubmit your project, this will override any previous submissions and only **the latest** submission will be considered – **resubmit at your own risk**.

#### Academic Integrity

The project you submit must be your own work. Minor consultations with others in the class are OK. The work on the project should be your own: all the design, programming, and testing should be done independently. Submissions that show excessive similarities will be taken as evidence of cheating and dealt with accordingly.