The goal of this project is to implement the generation of three-address code for expressions, as discussed in class. The result will be a compilable C program that uses the temporary variables from Project 3 and the three-address instructions discussed in class. Create a directory p4 under proj. Copy your Project 3 to p4 and start from there. Do all work for Project 4 in p4.

Goals

Starting point. First, comment out the call to astRoot.print(System.out) in class Compiler. Next, add a new call astRoot.genCode() immediately after astRoot.check(). The implementation of genCode() should print to System.out the generated three-address code.

Output. The output should be a complete C program that can be compiled with gcc. For example, if the input is

```
int e1() {
    int x;
    int y;
    int z;
    return x+y*z;
}
```

the output could be

```
int e1()
{
    int x;
    int y;
    int z;
    int _t2;
    int _t1;
    _t1 = y * z;
    _t2 = x + _t1;
    return _t2;
}
```

The grader will compile the produced code and, in some cases, will run it to ensure that it behaves correctly.

Input. The following restriction will be satisfied by the input program: the only statements in the program will be expression statements and return statements. You do not have to consider any other categories of statements. In the next project we will generalize to the full language.

Testing

Write many test cases and test your implementation with them. Submit at least 5 test cases with your submission. The test cases you submit will not affect your score for the project. Put them in the same location as the provided file t1.c and name them t2.c, ...
Submission

After completing your project, do

cd p4
make clean
cd ..
tar -cvzf p4.tar.gz p4

Then submit p4.tar.gz in Carmen.

General Rules (copied from the course syllabus)

1) Your submissions must be submitted electronically via Carmen by midnight on the due
date. The projects must compile and run on stdlinux. Some students prefer to implement
the projects on a different machine, and then port them to stdlinux. If you decide to use a different
machine, it is entirely your responsibility to make the code compile and run correctly on
stdlinux before the deadline. In the past many students have tried to port to stdlinux too close
to the deadline, leading to last-minute problems and missed deadlines.

2) Projects should be done independently. General high-level discussion of projects with other
students in the class is allowed, but you must do all design, programming, testing, and
debugging independently. Projects that show excessive similarities will be taken as evidence of
cheating and dealt with accordingly. Code plagiarism tools may be used to detect cheating. See
more details in the Syllabus under “Academic integrity”.

3) The projects are due by 11:59 pm on the due day. No exceptions will be made to this
deadline: if you submit at 12:00 am, your submission will be late. Please plan your time
carefully and do not submit in the last minute. You can submit up to 24 hours after the
deadline; if you do so, your project score will be reduced by 10%. If you submit more than 24
hours after the deadline, the submission will not be accepted, and you will receive zero points
for this project.

4) Accommodations for sickness and other special circumstances will be made based on
university guidelines. Please contact me ahead of time to arrange for such accommodations.