CSE 5343, Programming Project 5: Intermediate Code for Statements
Due Thursday, March 30, 11:59 pm (30 points)

The goal of this project is to implement the generation of three-address code for statements, as discussed in class. This includes the full language from earlier projects, including if-then, if-then-else, while-do, and for-loop. The result will be a compilable C program. The implementation of Project 4 (with necessary bug fixes) will be used to handle expressions. Create a directory $p5$ under $proj$. Copy your Project 4 to $p5$ and start from there. Do all work for Project 5 in $p5$.

Goals

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

```c
int e6() {
    int x;
    int y;
    x = 1;
    y = 2;
    if (x < y) return x;
    else return y;
}
```

output could be

```c
int e6()
{
    int x;
    int y;
    int _t1;
    x = 1;
    y = 2;
    _t1 = x < y;
    if (!t1) goto _l2;
    return x;
    goto _l1;
    _l2:
    return y;
    _l1:
};
```

This three-address code is based on the code generation rules described in class. Note that instruction `goto _l1;` is dead code; this is fine and you should not attempt to remove it.

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

End of program. At the very end of your output program, add an empty statement `;` as illustrated above. Depending on your input program, this empty statement may be redundant, or it may be necessary for the very last label you create.
**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, ...

Your submission must work correctly on test program lpc.c provided on the web page. *You can expect that a substantial number of points in the grading will be related to this test case.*

**Submission**

After completing your project, do
```
cd p5
make clean
cd ..
tar -cvzf p5.tar.gz p5
```

Then submit p5.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.