In homework 2, you are going to finish two tasks related control flow graph.

1. **(10 points)**   
   In the first task, you need to write a program to check whether the condition of an “if” statement contains an assignment statement. For example, **if (a = b) {}** is such a bug because **a = b** is an assignment statement but not a comparison statement.

You need to write a bug checker to find such kind of bugs in the given testing program. For each found bug, your program should print out a single line that contains the line number where such a bug is located. For example, for the testing program named ***t1\_testbug.c***attached here. You should print the following two lines, because there are two such bugs in the given program.

line 3  
line 11

1. **(15 points)**   
   In the second task, you need to write a program to generate CFGs in dot format for **if** statement and **for** **loop**. Clang already has commands to do this (checking this stackoverflow post for detail: <https://stackoverflow.com/questions/9225325/how-to-use-cfg-dump-option-with-clang-3-0>). Your program needs to simulate the **ViewCFG** command but in a simplified way. You only need to generate CFG for **if statement** and **for loop**.  
     
   You can refer to the output of **ViewCFG** to generate dot files in the correct format. It’s a little bit different with what you studied in class. If you are comfortable with what you studied in class, it’s also fine. In terms of the content in dot nodes, you can simply print the line numbers, or the statements themselves, whatever you are comfortable with.  
     
   The attached starting program (***t2\_cfg\_if.cpp***) already has the functionality to generate CFG for *single level* **if statement**. You don’t need to write your own code for **if statement**, but only for **for loop**. ***t2\_testcfg.png*** is CFG of ***t2\_testcfg.c***. You can use these files as a starting point. Be sure that you understand the logic in ***t2\_cfg\_if.cpp*** first, and then write your own code to generate CFG of **for loop**. You can assume that there is **no nested if statement and for loops**.
2. **Extra 5 points**   
   You got 5 extra points if you can generate CFG for nested if statements and for loops.
3. **What to submit**

Name your program for task 1 as **hw2\_t1.cpp**, and **hw2\_t2.cpp** for task2. Please follow this link to create a new Github repository for HW2, and then push your two .cpp files to Github. <https://classroom.github.com/a/_2FkcPBo>

**Some useful materials**:

1. There is a Makefile in attached files. It contains almost everything you need to compile your programs so you don’t need to write your own Makefile. If you decide to use it, please update the two paths in it: LLVM\_SRC\_PATH and LLVM\_BUILD\_PATH.
2. The attached example program used ASTConsumer to read ASTs. You can refer these two pages for details: <https://clang.llvm.org/doxygen/classclang_1_1ASTConsumer.html>
3. In this page, you will learn how to use RecursiveASTVisitor to find a specific type of AST nodes. <https://clang.llvm.org/docs/RAVFrontendAction.html>

http://clang.llvm.org/doxygen/classclang\_1\_1RecursiveASTVisitor.html

1. This Github repository contains a number of sample programs about using LLVM and Clang as libraries. <https://github.com/eliben/llvm-clang-samples>