**Project Bonus part hints:**

These are some hints on how to use LLVM's built in passes. You can use these as starting point and may have to explore from the LLVM/clang website.

Use the following command to generate LLVM IR from the C code:

**clang -emit-llvm -S multiply.c -o multiply.ll**

To convert LLVM IR to bitcode format, you need to use the following command:

**llvm-as multiply.ll -o multiply.bc**

Using the following command, we can convert LLVM bitcode to assembly code:

**llc multiply.bc -o multiply.s**

convert a bitcode file to an IR file:

**llvm-dis multiply.bc -o multiply.ll**

Use the opt tool to run the default passes supported in clang / llvm framework. Since most of the optimization is already done by the clang compiler, you may need to combine various optimizations and test for different machines to actually see the difference. You have to figure this out yourself.

Opt --help command should give all the details of passes.

The **opt** tool runs the transformation passes as:

**opt -mem2reg -S multiply.ll -o multiply1.ll**

**multiply1 is the optimized code.**