Homework 1

1.1 Generate optimal code (using 2 address or 3 address instructions) for the following loop. Assume all arrays have the same dimensions: (Note: FORTRAN stores arrays in column-major order.)

\[
\begin{align*}
d & i=1,n \\
& \quad do \ j=1,n \\
& \quad \quad k=k+1 \\
& \quad \quad a(i,k) = a(i,j) + c(i,j) \\
& \quad end \ do \\
& end \ do.
\end{align*}
\]

1.2 Construct the dag for the following basic block

1. d = b * c
2. e = a + b
3. b = b * c
4. a = e - d

What are the legal evaluation orders assuming that a, b, and c are live at the end of the basic block? How about if only b is live at the end?

1.3 Modify the gencode algorithm for a target machine whose arithmetic operation instructions have three operands, all in registers. The original gencode algorithm produces optimal code. Is your algorithm also optimal? Why or why not?

1.4 Let \( T \) be an expression tree whose root is labeled \( N \geq 2 \) after applying the labeling algorithm we used for gencode. Show that \( T \) contains at least \( 3 \times 2^{N-2} - 1 \) interior nodes.

2.1 For the given routine, generate the control flow graph and then show the interval derived sequence.

2.2 Write an irreducible program that makes sense.

2.3 Is the following flow graph reducible or irreducible? Compute the interval derived sequence to prove your statement.

2.4 Design an efficient algorithm to construct the dominance tree of a reducible flow graph.
2.5 Design an algorithm to identify IF statements in a control flow graph. Your algorithm should deal properly with all sort of branching situations and with nested IF statements. Here IF statements means the compound construct containing both the IF condition and all the statements whose execution is controlled by the condition.