Question 1. Consider the compilation of the following (miniJava, Java, C, ...) expression for a register-oriented processor architecture like MIPS (here variables $x$ and $y$ are integer variables stored in memory)

$$(x + y) \times x + (x + y)$$

(a) Show an AST for this expression and use Sethi-Ullman numbering on each node to determine the minimum number of registers needed to evaluate the expression. How many registers are needed?
(b) Show the tuple code generated by simple code generation using temporaries.

(c) Show how the tuple code in (b) can be simplified to the code in (d) (next page) using common subexpression elimination.
(d) Determine the lifetimes of each temporary $t_1 - t_5$.

\[
\begin{array}{ccccc}
 & t_1 & t_2 & t_3 & t_4 & t_5 \\
t_1 &=& x & & & \\
t_2 &=& y & & & \\
t_3 &=& t_1 + t_2 & & & \\
t_4 &=& t_3 \times t_1 & & & \\
t_5 &=& t_4 + t_3 & & & \\
\end{array}
\]

(e) What is the minimum number of registers $k$ needed to evaluate the tuple code above? Show an assignment of temporaries to $k$ registers by constructing the interference graph and coloring it using $k$ colors.
Problem 2: Dataflow analysis

Consider the following program fragment (never mind that the program is stupid).

```java
while (x > z) {
    x = (x - z) * 3;
    z = x + 5;
}
y = x;
```

We construct the control flow graph below for this program fragment which has variables \{x, y, z, t0, t1, t2, t3\}. Write equations \(L_i\) for variables live at each program point \(i \in \{1, \ldots, 11\}\). Solve the equations by starting from an initial value for the \(L_i\) and iterate till a fixpoint is reached. Using the final values of the \(L_i\), construct the interference graph for the program variables on the next page. How many registers are needed to run this program fragment?