Some of the following exercises ask you to generate target code. Use the instructions and addressing modes of the machine models that were discussed in the lecture.

**Exercise 1 (Register Allocation with Usage Counters)** Fig. 1 shows a control flow graph for a loop with six basic blocks. For simplicity, the branch instructions are neglected in this example. There are three registers R0, R1 and R2 available. Two of the registers, R0 and R1, are to be used for frequently used loop variables. Active variables in R2 must be saved at the end of each basic block.

![Control Flow Graph](image)

- Determine the two variables that should be assigned to R0 and R1.
- Generate the target code sequences for the individual basic blocks.
Exercise 2 (Register Allocation by Graph Coloring) The following basic block forms the body of a loop. At the end of the basic block the variables e and f are active.

(1) t1 := f + e
(2) t2 := e - 10
(3) t3 := t2 + f
(4) t4 := t1 - e
(5) t1 := t3 * t4
(6) t5 := t1 + 20
(7) f := t5 + t4
(8) e := t5 * t4

• Analyze the life times of the variables and construct the corresponding register conflict graph.
• Determine the register allocation without register spills (temporarily saving a variable in memory) for four available registers R0, R1, R2, R3.
• Can you find a register allocation without register spills if there are only three registers available?
• In case a register spill is unavoidable, a variable must be selected that is saved to memory and, later on, reloaded into a register. Which strategies for this selection process can you imagine?

Exercise 3 (Code Generation for DAGs) Given is the following basic block. Only variable d must be active at the exit.

a = v * s
b = t - u
c = t - a
d = (r * s + b * c) - (b * c + b * t * t)

• Construct the DAG for the basic block.
• Number the nodes of the DAG and determine a computation schedule (order) for the nodes. Apply the heuristics discussed in the lecture.
• Generate the corresponding 3-address code.

Exercise 4 (Code Generation by Dynamic Programming) Generate optimal code for the following expression under the assumption that you have three registers available. Use the code generation procedure based on dynamic programming.

(a - b * c) + (d * e - g * h)