1. Java programs (byte code) can be executed (at least) in four different ways: interpretation, ordinary compilation, Just-In-Time (JIT) compilation, or directly on a Java-processor.
   a. Give examples on a situation where ordinary compilation would be better than the alternatives. Explain, why each alternative would be worse.
   b. Give similar examples and explanations on when the other alternatives would be best choice.
   c. What would be best alternative to execute (Java) programming project at school? Why?
   d. What processes are needed at execution time, when your Java program MyProg is executed with 1) interpretation, 2) ordinary compilation, 3) JIT compilation, 4) Java processor?
   e. When would you use C# instead of java? What do C# and Java have in common? How do they differ?
   f. What kind of programming language is Scala? What does it have to do with Java?

2. Subroutines vs. macros, literals vs. constants
   a. Give an example on a situation where routine XYZ would be better to implement as a macro instead of as a subroutine. Explain. Give an example?
   b. Give an example on a situation where routine XYZ would be better to implement as a subroutine instead of as a macro. Explain. Give an example?
   c. Give an example on a situation where value X would be better to implement as a literal instead of a constant in an instruction. Explain. Give an example?
   d. Give an example on a situation where value X would be better to implement as a constant in instruction instead of a literal. Explain. Give an example?

3. Literals, variables, parameters
   a. How does using a literal differ from using a (global) variable? Advantages? Disadvantages?
   b. What danger is there if one could write to literal area?
   c. How does using a literal differ from using the constant field in an instruction?
   d. Macros use usually call-by-name parameters instead of call-by-value or call-by-reference parameters. Could you use call-by-value or call-by-reference parameters in macros?

4. What does program mystery.k91 do? How does it do it?
   And what about program mystery3.k91? How does it do it?
   How could you use this type of programming?
   What problems are there with this type of programming?

5. Assume that your are encoding a ttk-91 simulator with Titokone. Your simulator reads machine language ttk-91 code and emulates the execution with simulated instructions, one instruction at a time.
   a. How would you define the simulated ttk-91 structures (registers, memory) in your program?
   b. How would you code (with which ttk-91 instructions) the fetch-phase of the fetch-execute cycle?
   c. How would you code breaking one machine instruction into its fields? (Hint: use bit masks with and, shl, shr-instructions)
   d. How would you implement fetching the second operand to TR?
   e. How would you code the execution phase of instruction "add r2,r3"?
      You may assume that the second operand value is already in TR.
   f. How would you code the execution phase of instruction "jump loop"?
      You may assume that the second operand value is already in TR.

6. Answer the Class Feedback. It would be nice, if in question 7 you would give comments on the usability and functionality of the mini exams in their current form.
   a. Did you take the mini exams or not?
   b. If you did take part of them, did you find any advantage on them? What?
   c. If you did not take them, why not?
   d. Did you answer in the course exam also to other questions but question 4 (mini exam 4)? Why?
e. In which ways could the current mini exams be made better?
f. Do you think that mini exams are a bad idea, and that we should get rid of them as quickly as possible? Why?