

Project 2: Program analyses, code transformations, and improved code generation — assigned incrementally and finally due on Friday 5 December — final presentations on Thursday 11 December (the official final exam day)

This first incremental release consists of Tasks 1–2. Release date 30 October. Submissions due 5 November.

Project 2 is a continuation of Project 1. You will now implement an optimizing compiler for the language WHILE.

Task 1: Consolidate all work from Project 1. (Tidy up and fix any problems you may have noticed.)

2.1 Labelled syntax

The labelled syntax decorates computational blocks (“program points”) with labels ℓ :

Commands $c ::= [x := a]^\ell \mid [\text{skip}]^\ell \mid c_1; c_2 \mid \text{if } [b]^\ell \text{ then } c_1 \text{ else } c_2 \text{ fi} \mid \text{while } [b]^\ell \text{ do } c_1 \text{ od}$

Task 2: Given an AST, produce a decorated AST with unique labels (conventionally, we use distinct numbers as labels). Display the decorated AST. Display (pretty-print) the annotated program in source form (with labels inside comments inserted at the appropriate place).

What to turn in

Submit a tar file containing: sample While programs; the same programs after labelling; and the corresponding decorated ASTs (PDF).