

## Translator Design - Lab 2

1. Consider the exponentiation function from `examples/func.imp`. Unfortunately, variable declarations are not yet supported. Rewrite `exp` without the use of variables, ensuring the time complexity is unchanged.
2. The function now relies on `if` statements - add the keywords used in `if` statements to the lexer.
3. Define an AST node to represent `if` statements, ensuring the presence of an optional alternative branch.
4. Compile `if` statements to bytecode.
  - (a) Identify the number of labels required and allocate them.
  - (b) Lower the condition check and emit a conditional jump.
  - (c) Recursively compile the bodies of the branches.
  - (d) How would the `if` statement be lowered if there was a `JUMP_TRUE` opcode?