

# Homework 8

Sunday, November 24, 2019

2:32 PM

Q1: Write a program in the WHILE-language that calculates the factorial function.

A1:

```
write "Input";
read n;
res := 1;
while n != 0 do {
    res := res * n;
    n := n - 1
};
write "Result:";
write res
```

Q2: What optimisations could a compiler perform when compiling a WHILE program?

A2:

- Replacing  $+1/-1$  with increment/decrement operations.
- With respect to the previous case with the factorial program, if the  $n$  would be known in advance, a compiler could unroll the loop.

Q3: What is the main difference between the Java assembler (as processed by Jasmin) and Java Byte Code?

A3: The assembly language is in a plain text format and can still be read by humans, whereas the Java Byte Code

is in a binary format and cannot be reasonably understood by humans. Other minor differences include absence of labels in byte code and also absence of comments.

Q4: Parser combinators can directly be given a string as input, without the need of a lexer. What are the advantages to first lex a string and then feed a sequence of tokens as input to the parser?

A4: By lexing a string, we can preprocess it and remove redundant information like comments and whitespace, which means our parser syntax becomes a lot simpler.

Q5: Explain what is meant by the terms lazy evaluation and eager evaluation.

A5: Lazy evaluation means evaluation the arguments to a function only when the body of the function immediately needs them to perform some computation. Eager evaluation on the other hand evaluates all function arguments before even starting to work on the function body.