Homework 8

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

2. What optimisations could a compiler perform when compiling a WHILE-program?

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

4. Consider the following Scala snippet. Are the two functions `is_even` and `is_odd` tail-recursive?

```scala
def is_even(n: Int) : Boolean = {
    if (n == 0) true else is_odd(n - 1)
}

def is_odd(n: Int) : Boolean = {
    if (n == 0) false
    else if (n == 1) true else is_even(n - 1)
}
```

   Do they cause stack-overflows when compiled to the JVM (for example by Scala)?

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

6. **(Optional)** This question is for you to provide regular feedback to me: for example what were the most interesting, least interesting, or confusing parts in this lecture? Any problems with my Scala code? Please feel free to share any other questions or concerns. Also, all my material is ~~crap~~ imperfect. If you have any suggestions for improvement, I am very grateful to hear.

   If *you* want to share anything (code, videos, links), you are encouraged to do so. Just drop me an email.