

CvP - Werkcollege 11

Exercise 1 Rewrite the following scheme function to a Haskell function:

```
(define factorial
  (lambda (x)
    (if (eq? x 1)
        1
        (* x (factorial (- x 1)))
    )
  )
)
```

Exercise 2 Using pattern matching, find a function definition of head (car in scheme) and of tail (cdr in scheme).

Exercise 3 Create a function that takes a list, removes even numbers, and replaces them with "BOOM" if they are lower than 10, and by "BANG" otherwise. You can use the function odd which returns true if a number is odd.

Exercise 4 Create an endless list of all numbers divisible by 7, greater than 71 when they are doubled.

Solution 5 Create an endless list of all Fibonacci numbers using cons, zipWidth and tail.