Problems Solved:

16	17	18	19	20

Name:

Matrikel-Nr.:

Problem 16. Construct a Turing machine $M = (Q, \Gamma, \sqcup, \{0,1\}, \delta, q_0, F)$ such that $L(M) = \{1^k 0 1^{k+1} \mid k \in \mathbb{N}\}$. Write down Q, Γ, F and δ explicitly.

Problem 17. Write a RAM program that from a given natural number n prints its binary representation. In order to simplify the problem the output shall be in low positions first format, i.e., the number 8_{10} is 0001_2 but not 1000_2.
Hint: please note that the computation of the quotient respectively remainder of a division by 2 can be implemented by the repeated subtraction of 2.

Problem 18.

1. Show by using *only* the Definition of a *loop program* (Def. 23 in the lecture notes, Section 3.2.2) that the function

$$s(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 < x_2, \\ 0 & \text{otherwise} \end{cases}$$

   is loop computable. I.e. give an explicit loop program for s.

   Note that it is not allowed to use an abbreviation like

```
xi := xj - xk;
```

2. Write a loop program that computes the function $d : \mathbb{N} \to \mathbb{N}$ where $d(x_1, x_2)$ is $k \in \mathbb{N}$ such that $k \cdot (x_2 + 1) = x_1 + 1$ if such a k exists. The result is $d(x_1, x_2) = 0$, if a k with the above property does not exist.

   For simplicity in the program for d, you are allowed to use a construction like the following (with the obvious semantics) where P is an arbitrary loop program.

```
IF xi < xj THEN P END;
```

Problem 19. Provide a loop program that computes the function $f(n) = \sum_{k=1}^{n} k(k+1)$, and thus show that f is loop computable.
You are only allowed to use the constructs given in Definition 23 of the lecture notes.

Problem 20. Suppose P is a while-program that does not contain any `WHILE` statements, but might modify the values of the variables x_1 and x_2.
Transform the following program into Kleene's normal form.
Hint: first translate the program into a goto program, replace the `GOTO`s by assignments to a control variable, and add the `WHILE` wrapper.

```
x0 := 0
WHILE x1 DO
    x1 := x1 - 1;
    x2 := x1;
    WHILE x2 DO
```

```
            P;
      END;
END;
x0 := x0 + 1
```