Problems Solved:

16	17	18	19	20

Name:

Matrikel-Nr.:

Problem 16.

Definition 1 (RAM computable)**.** We say that a partial function $f : \mathbb{N} \to_P \mathbb{N}$ is *RAM computable* if there exists a RAM R that such that

- R terminates for input $n \in \mathbb{N}$ if and only if $n \in domain(f)$;
- R terminates for input $n \in \mathbb{N}$ with output n' if and only if $n' = f(n)$.

Show that every loop computable function is also RAM computable by describing how the loop program computing the function can be translated to a RAM program.

Problem 17. Give reasons for your answers.

1. Let R be a RAM that reads exactly one number from its input tape and always terminates with 0 or 1 written on its output tape. Is there a function $f : \mathbb{Z} \to \mathbb{Z}$ such that $f(x) = y$ if and only if the input was x and after termination y is on the output tape of R?
2. Let $f : \mathbb{Z} \to \mathbb{Z}$ be a function. Is there always a RAM R such that R terminates on every input and that R with input $z \in \mathbb{Z}$ has written $f(z)$ to its output tape?

Problem 18. 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.

Problem 19. Write down explicit loop programs for s and d.

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 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
```