**Problems Solved:**

| 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|

**Name:**

**Matrikel-Nr.:**

**Problem 21.** Let $f$ be a primitive recursive function defined by the recursive equations

$$f(0, y) = 2, \quad f(x+1, y) = f(x, y)^y$$

1. Compute $f(3, 3)$.
2. Show that $f$ is indeed a primitive recursive function by defining it explicitly from the base functions, the (primitive recursive) function $\varepsilon(x, y) = x^y$, composition, and the primitive recursion scheme.

**Problem 22.**

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

   $x_i := x_j - x_k$;

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** $x_i < x_j$ **THEN** P **END**;

**Problem 23.** Let $S : \mathbb{N} \to \{0, 1\}$ be defined by

$$S(x) = \begin{cases} 1 & \text{if } x \text{ is the square of some natural number,} \\ 0 & \text{otherwise.} \end{cases}$$

1. Write a logical formula for the statement "x is the square of some natural number" used in the definition of $S$.
2. Show that $S$ is primitive recursive by giving a primitive recursive definition of $S$.
3. Show that $S$ is loop computable by giving a loop program that computes $S$.

*Hint:* It is OK to assume that the equality check and multiplication are primitive recursive as well as loop computable. The equality check $e$ is given by $e(x, y) = 1$ if $x = y$ and $e(x, y) = 0$ otherwise. Any predicate $P$ can be encoded as a function $f$. We define $f(x) := 1$ if $P(x)$ is true and $f(x) := 0$ if $P(x)$ is false.

**Problem 24.** Let $P$ be the following program for counting how many of the first $n$ odd numbers starting with 3 are prime.

```
s :=0
i :=3
LOOP n DO
    p = isprime(i)
      IF p = 1 THEN s := s+1
      END;
    i := i+2
END;
```

Convert $P$ into primitive recursive function, provided *isprim* is a given loop program (you may assume that a corresponding primitive recursive function *isprim* is given as well).

**Problem 25.** Let $p$, $s$, and $f$ be defined as:

$$p(x) := \mu y.s(x, y) = 1$$

$$s(x, 0) := x$$
$$s(x, y + 1) := f(s(x, y))$$

$$f(n) = \begin{cases} 3n + 1 & \text{if } n \text{ is odd,} \\ \frac{n}{2} & \text{if } n \text{ is even.} \end{cases}$$

1. Provide an explicite computation of $p(3)$.
2. Construct a while program that computes $p$.
3. Based on what you read (e.g. in Wikipedia) about the Collatz-Problem (or Collatz conjecture), do you think that $p$ is primitive recursive or not? Justify your answer.