Problems Solved:

21	22	23	24	25

Name:

Matrikel-Nr.:

Problem 21. Let $f : \mathbb{N} \to \mathbb{N}$ be the (partial) function

$$f(x) = \begin{cases} y & \text{such that } x = y^2 \text{ if such a } y \text{ exists,} \\ \text{undefined} & \text{if there is no } y \text{ with } x = y^2. \end{cases}$$

1. Is f loop computable? (Justify your answer.) If your answer is “yes”, define f by a LOOP program. Here you are also allowed to use an *if-then-else*-like statement.
2. Is f while computable? (Justify your answer.) If your answer is here “yes” but your answer to 1 was “no”, define f by a WHILE program where you are allowed to use the same constructions as in 1.
3. Is f primitive recursive? If your answer is “yes”, define f by using the base functions, composition and the primitive recursion scheme Additionally you are allowed to use the (primitive recursive) functions

$$m : \mathbb{N}^2 \to \mathbb{N}, \quad (x, y) \mapsto x \cdot y$$

$u : \mathbb{N}^2 \to \mathbb{N}$,

$$u(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

and $IF : \mathbb{N}^3 \to \mathbb{N}$,

$$IF(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise.} \end{cases}$$

Other functions or rules are forbidden.

4. Is f μ-recursive? If your answer is here “yes” but your answer to 3 was “no”, define f as described in 3 with the additional construction of μ-recursion. Is your construction in Kleene’s normal form? If it is not, describe an (informal) procedure how one can turn it into Kleene’s normal form.

Problem 22. Let $f : \mathbb{N} \to \mathbb{N}$ be the function

$$f(x) = \begin{cases} y & \text{such that } x = y^2 \text{ if such a } y \text{ exists,} \\ 0 & \text{if there is no } y \text{ with } x = y^2. \end{cases}$$

1. Is f loop computable? (Justify your answer.) If your answer is “yes”, define f by a LOOP program. Here you are also allowed to use an *if-then-else*-like statement.

2. Is f while computable? (Justify your answer.) If your answer is here "yes" but your answer to 1 was "no", define f by a WHILE program where you are allowed to use the same constructions as in 1.

3. Is f primitive recursive? If your answer is "yes", define f by using the base functions, composition and the primitive recursion scheme Additionally you are allowed to use the (primitive recursive) functions

$$m : \mathbb{N}^2 \to \mathbb{N}, \quad (x, y) \mapsto x \cdot y$$

$u : \mathbb{N}^2 \to \mathbb{N}$,

$$u(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

and $IF : \mathbb{N}^3 \to \mathbb{N}$,

$$IF(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise.} \end{cases}$$

Other functions or rules are forbidden.

4. Is f μ-recursive? If your answer is here "yes" but your answer to 3 was "no", define f as described in 3 with the additional construction of μ-recursion. Is your construction in Kleene's normal form? If it is not, describe an (informal) procedure how one can turn it into Kleene's normal form.

Problem 23. 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.

   Note that according to Definition 29 (lecture notes), in the composition scheme the g_i have the same number of arguments as the h. Similarly, in the primitive recursion scheme, recursion is done on the first argument of h and the respective f has one argument less while g has one argument more than h. You are not allowed to deviate from these formal requirements.

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 *isprime* is a given loop program (you may assume that a corresponding primitive recursive function *isprime* is given as well).

Problem 25. Let $q : \mathbb{N}^2 \to \mathbb{N}$, $(x, y) \mapsto x \cdot x$ (sic!) and $u : \mathbb{N}^2 \to \mathbb{N}$,

$$u(x,y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y, \end{cases}$$

be given primitive recursive functions.
Let $r : \mathbb{N}^2 \to \mathbb{N}$ be definied by

$$\begin{aligned} r(x) &= (\mu p)(x) && \text{minimization} \\ p(y,x) &= u(q(y,x), \mathrm{proj}_2^2(y,x)) && \text{composition} \end{aligned}$$

Informally we have

$$r(x) = \min_y \{ y \in \mathbb{N} \mid u(q(y,x),x)) = 0 \}$$

Similar to the treatise in the lecture notes, construct a while program that computes r. For simplicity, you are allowed to write statements such as $x_k = q(x_i, x_j)$ and $x_k = u(x_i, x_j)$ into your program. What will your program compute if it is started with input $x_1 = 2$?