Problems Solved:

21	22	23	24	25

Name:

Matrikel-Nr.:

Problem 21. We know that the function $p : \mathbb{N}^3 \to \mathbb{N}$ defined by $p(a, b, n) = a \uparrow^n b$ is not primitive recursive. However, the function $p_2 : \mathbb{N}^2 \to \mathbb{N}$, defined by $p_2(a, b) = a \uparrow^2 b$ **is** primitive recursive.
Show that fact by defining p_2 explicitly from the base functions, the (primitive recursive) function $\varepsilon(x, y) = x^y$, composition, and the primitive recursion scheme.

Problem 22. 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.)
2. Is f a primitive recursive function? (Justify your answer.)
3. Define f by using the base functions, composition, the primitive recursion scheme, and μ-recursion. Additionally you are allowed to use the (primitive recursive) functions

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

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

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

4. Why do you need the μ-recursion in your construction?
5. Is you 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 P be the following program.

```
x0 := x1 + 1
LOOP x1 DO
    LOOP x0 DO
        x0 := x0 + 1;
    END;
END;
```

Similar to the construction in the lecture notes, let $f_P : \mathbb{N}^2 \to \mathbb{N}^2$ be the function that maps the given $(0, x_1)$ at the start of P to the values (x_0, x_1) after the execution of the program P. Show that f_P is primitive recursive by translating the loop program into a primitive recursive definition for f_P. Follow the steps given in the lecture notes.
Compute $f_P(0, 1)$ via your primitive recursive definition and compare it with the result you get from executing P with input $x_1 = 1$.

Problem 24. 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

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

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 $x_1 = 2$?

Problem 25. Let f be defined as

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

Now one can imagine the following process. Choose a positive natural number and apply f to it. If the result is 1 then terminate the process, otherwise apply f again and iterate the process until the result is 1.
Let ν be the function that takes a positive natural number x as input and returns the number of iterations in the above process until its termination, i.e. if the process terminates then

$$f^{\nu(x)}(x) = \underbrace{f(f(\cdots f(}_{\nu(x)\text{-fold}} x) \cdots) = 1.$$

Show that ν is a recursive function. You may use any theorems from the lecture notes and you can use the (primitive recursive) function $\text{pred} : \mathbb{N} \to \mathbb{N}$,

$$\text{pred}(x) = \begin{cases} 0 & \text{if } x = 0, \\ x - 1 & \text{if } x > 0. \end{cases}$$

Bonus task (very difficult): Is ν primitive recursive?