Exercise 1 (5 Points, Weakest precondition)

Calculate the weakest precondition of the following program with respect to the postcondition \( \{5 \leq y < 23\}\).

\[
x := 4y;
\text{if (}x \leq 12\text{)} \{
    y := 2x + 5
\} \text{ else } \{
    y := 3x - 2
\};
y := y + 4
\]

Exercise 2 (10 Points, Partial correctness)

Consider the following programs and use the annotation calculus to prove that the program is partially correct regarding the given pre and postconditions (note, that also the proof obligations have to be verified).

(a) \text{if (}z > y\text{)} \{
    x := y;
\} \text{ else } \{
    x := z;
\}

Precondition \( \top \), postcondition \( x = \min\{y, z\}\).

(b) \text{a := 0;}
\text{z := 1;}
\text{while (a < x) } \{
    \text{z := z * }y;
    \text{a := a + 1;}
\}

Precondition \( x \geq 0 \), postcondition \( z = y^x \).

Exercise 3 (10 Points, Total correctness)

Consider the following program \text{Mult}:

\[
a := 0;
z := 0;
\text{while (a \neq y) } \{
    z := z + x;
    a := a + 1
\}
\]
Use the annotation calculus to prove that \( \vdash t \{ y \geq 0 \} \text{Mult}\{ z = xy \} \)

**Exercise 4 (10 Points, Invariant)**

Determine a suitable invariant for the program and show, using the annotation calculus, that it is partially correct under the precondition \( x = x_0 \land x \geq 0 \) and the postcondition \( x = \frac{x_0(x_0+1)}{2} \). **Explain your invariant.**

\[
\begin{align*}
z &:= x; \\
\text{while } z > 0 \{ \\
& \quad z := z - 1; \\
& \quad x := x + z; \\
\} \\
\end{align*}
\]

(Do not forget the proof obligations!)

**Exercise 5 (0 Points, Extra (15P): A tricky algorithm)**

Consider the following program \( P \) (it uses \( x \) as input and produces the output \( y \)):

\[
\begin{align*}
i &:= 0; \\
y &:= 0; \\
\text{while } (i \neq x) \{ \\
& \quad i := i + 1; \\
& \quad y := y + 2i \\
& \quad y := y - 1 \\
\} \\
\end{align*}
\]

a) Which function \( f : \mathbb{N} \rightarrow \mathbb{N} \) does \( P \) compute?

b) Formulate appropriate pre and postconditions and prove total correctness using the annotation calculus.