1) $f_1(x) = 5x + 6$

The simplest way is to multiply by 5, then add 6.

Instead, we implement $f_1(x) = 5(x+1) + 1$

$$\begin{array}{c}
\text{Input: } x \\
\text{Output: } 5x + 6
\end{array}$$

2) $f_2(x, y) = 3x + y$, for every $x, y \geq 0$

   a) Compute $3x$

   b) If $y > 0$:
      Subtract 1 from $y$
      Add 7 to $3x$ over
      When $y = 0$, the result is in $3x$ over.

   $\begin{array}{c}
   \text{Input: } x, y \\
   \text{Output: } 3x + y
   \end{array}$

3) $f_3(x)$

   If you are multiplying $y$ by 7, you first need to shift $y$ to the right by 3 positions.
Out of laziness, I am implementing a simple algorithm suggested by one of the students in the class. I want to emphasize that any algorithm can be implemented.

Given \( x_1, x_2, \ldots, x_n \), we first transform it to \( 0, u, L, x_1, u, L, \ldots, u, L, x_n \).

Then repeatedly subtract 1 from each \( x_i \) and add 1 to the result until one of the \( x_i \)'s becomes 0. Then the result contains the min value. Then trace the \( x_i \)'s.

This doesn't require that all the input numbers have the same number of bits.