Exercise 1: (H) Generating Loop Bounds

Assume we want to reverse the loop-nesting orders in the following code:

```c
for (i = 0; i <= 5; i++)
    for (j = i; j <= 7; j++)
        A[j][i] = 0;
```

We first write down all the constraints from the two loops:

\[
\begin{align*}
0 &\leq i \\
0 &\leq j \\
i &\leq 5 \\
j &\leq 7
\end{align*}
\]

In our new iteration order, the innermost variable is \( i \), so we infer all the constraints on \( i \):

\[
\begin{align*}
0 &\leq i \\
i &\leq 5 \\
i &\leq j
\end{align*}
\]

This gives us the code for the innermost loop:

```c
for (i = 0; i <= min(5,j); i++)
```

Now, we perform Fourier-Motzkin elimination to get rid of the variable \( i \). We obtain one additional constraint, \( 0 \leq j \), which replaces all inequalities involving \( i \) in the original set. We end up with two constraints, \( 0 \leq j \) and \( j \leq 7 \), yielding the following code:

```c
for (j = 0; j <= 7; j++)
    for (i = 0; i <= min(5,j); i++)
        A[j][i] = 0;
```

We’re done, except you haven’t done anything yet, so please perform the precise same transformation on the following loop:

```c
for (i = 1; i <= 30; i++)
    for (j = i+2; j <= 40-i; j++)
        A[j][i] = 0;
```
Exercise 2: (H) Deforestation

Let \texttt{comp}, \texttt{id}, \texttt{map}, \texttt{foldl}, and \texttt{tabulate} be defined as in the lecture. Given the following code:

\begin{verbatim}
let sq x = x * x
let sumSquares = comp (foldl (+) 0) (comp (map sq) (tabulate id))
\end{verbatim}

Simplify the function \texttt{sumSquares} as much as possible.