The first step in translating from $P_0$ to x86 is to flatten complex expressions into a series of assignment statements. For example, the program

$$\text{print} \ - \ \text{input()} + 2$$

is translated to the following

$$\begin{align*}
tmp0 &= \text{input()} \\
tmp1 &= -tmp0 \\
tmp2 &= tmp1 + 2 \\
\text{print} \ tmp2
\end{align*}$$

In the resulting code, the operands of an expression are either variables or constants, that is, they are simple expressions. If an expression has any other kind of operand, then it is a complex expression.

**Exercise 1.3.** Write a recursive function that flattens a $P_0$ program into an equivalent $P_0$ program that contains no complex expressions. Test that you have not changed the semantics of the program by writing a function that prints the resulting program. Run the program using the standard python interpreter to verify that it gives the the same answers for all of your test cases.