Solution to Problem 1

**Problem 1.** Prove that the function computing the sum $1^2 + 2^2 + 3^2 + \ldots + n^2$ is primitive recursive. This proof should follow the same pattern that we used in class to prove that addition and multiplication are primitive recursive:

- You start with a 3-dot expression.
- First you write a for-loop corresponding to this function
- Then you describe this for-loop in mathematical terms
- Then, to prepare for a match with the general expression for primitive recursion, you rename the function to $f$ and the parameters to $n_1, \ldots, n_k, m$
- Then you write down the general expression of primitive recursion for the corresponding $k$
- Then you match: find $f$ and $g$ for which the specific case of primitive recursion will be exactly the functions corresponding to initialization and to what is happening inside the loop
- Then, you get a final expression for the function $1^2 + 2^2 + 3^2 + \ldots + n^2$ that proves that this function is primitive recursive, i.e., that it can be formed from $0, \pi_k^i, \text{ and } \sigma$ by composition and primitive recursion.

**Solution.** Here is the for-loop for computing the desired expression:

```c
int sum = 0;
for (int i = 1; i <= m; i++) {
    sum = c + i * i;
}
```

Let us now describe this for-loop in mathematical terms. After iteration $i$, we add the number $i + 1$. The value $\text{sum}(i)$ is the value of the variable $\text{sum}$ after iteration $i$. So, $\text{sum}(m+1)$ is the value of the variable $\text{sum}$ after iteration $m+1$. In this case, $i = m + 1$, so we get:

$$\text{sum}(0) = 0;$$
$$\text{sum}(m + 1) = \text{sum}(m) + (m + 1) \cdot (m + 1).$$
To prepare for the match, we rename the function to \( h \) (here, there are no other parameters to rename):

\[
\begin{align*}
  h(0) &= 0; \\
  h(m + 1) &= h(m) + (m + 1) \cdot (m + 1).
\end{align*}
\]

Here, we are defining a function of 1 variable. In general, primitive recursion defines a function of \( k + 1 \) variables. Here, \( k + 1 = 1 \), so \( k = 0 \), and the general expression for primitive recursion takes the following form:

\[
\begin{align*}
  h(0) &= f; \\
  h(m + 1) &= g(m, h(m)).
\end{align*}
\]

To match with the above description, we need to take \( f = 0 \) and \( g(m, h) = h + (m + 1) \cdot (m + 1) \), i.e., \( g = add(\pi_2^2, prod(\sigma \circ \pi_1^2, \sigma \circ \pi_1^2)) \).

Thus, the desired expression for our function is

\[
sum = PR(0, add(\pi_2^2, prod(\sigma \circ \pi_1^2, \sigma \circ \pi_1^2))).
\]