Solution to Problem 1

Problem 1. Prove that the function computing the sum $1 + 2 + \ldots + n$ 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 g and h 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 $1+2+\ldots+n$ that proves that this function is primitive recursive, i.e., that it can be formed from 0, π_i^k, and σ by composition and primitive recursion.

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

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

Let us now describe this for-loop in mathematical terms:

$$sum(0) = 0;$$

$$sum(m + 1) = sum(m) + (m + 1).$$

To prepare for the match, we rename the function to h (here, there are no other parameters to rename):

$$h(0) = 0;$$

$$h(m+1) = h(m) + (m+1).$$

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:

$$h(0) = f;$$

$$h(m+1) = g(m, h(m)).$$

To match with the above description, we need to take $f = 0$ and $g(m, h) = h + (m+1)$, i.e., $g = add(\pi^2_2, \sigma \circ \pi^2_1)$.

Thus, the desired expression for our function is

$$sum = PR(0, add(\pi^2_2, \sigma \circ \pi^2_1))$$