Solution to Problem 7

Problem. Prove that the following function is mu-recursive:

```c
int j = 2;
while(!(a + b * j <= c))
  {j++;}
```

Solution. According to the general algorithm, first, we write a similar problem, but with a for-loop:

```c
int j = 2;
for(i = 1; i <= m; i++)
  {j++;}
```

This program can be translated into primitive recursion as follows:

\[ j(0) = 2; \]
\[ j(m + 1) = j(m) + 1. \]

A general primitive recursion defines a function \( h(n_1, \ldots, n_k, m) \) of \( k + 1 \) variables. In our case, we have a function of 1 variable, so \( k + 1 = 1 \) and \( k = 0 \). For \( k = 0 \), the general primitive recursion has the form

\[ j(0) = f; \]
\[ j(m + 1) = h(m, j(m)). \]

Here, \( f(0) = 2 = \sigma(\sigma(0)) \) and \( h = \sigma \circ \pi_2^2 \), so

\[ j(m) = PR(\sigma(\sigma(0)), \sigma \circ \pi_2^2). \]

As the number of iterations, we take the smallest \( m \) for which \( a + b \cdot j(m) \leq c \), i.e., \( \mu m.(a + b \cdot j(m) \leq c) \). Thus, the desired function has the form

\[ j(\mu m.(a + b \cdot j(m) \leq c)). \]