Solution to Problem 7

**Problem.** That the following function is mu-recursive:

```c
int j = 1;
while(!(a * j <= m))
    {j++;}
```

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

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

This program can be translated into primitive recursion as follows:

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

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

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

Here, \(f(0) = 0\) and \(h = \sigma \circ \pi_2^3\), so

\[
j(m) = PR(0, \sigma \circ \pi_2^3).
\]

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

\[
j(\mu n.(a \cdot j(n) \leq m)).
\]