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

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

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

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

This program can be translated into primitive recursion as follows:

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

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 2 variables, so \( k + 1 = 2 \) and \( k = 1 \).

For \( k = 1 \), the general primitive recursion has the form

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

Here, \( f(a) = a = \pi_1^1 \) and \( g = \sigma \circ \pi_3^3 \), so

\[
j(m) = PR(\pi_1^1, \sigma \circ \pi_3^3).
\]

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

\[
j(a, \mu m.!(j(a, m) * j(a, m) \leq b)).
\]