Task: Apply the general algorithm for transforming the finite automaton into a regular language (i.e., a language described by a regular expression) to Automaton from Problem 1.1. For simplicity, assume that we only have symbols $A$, $a$, and $1$. Eliminate first the error state, then the start state, and finally, the state $n$.

Solution. We start with the described automaton:

According to the general algorithm, first we add a new start state $ns$ and a few final state $nf$, and we add jumps:

- from the old start state to the new start state, and
- from each old final state to the new final state.

As a result, we get the following automaton:
Eliminating the sink state. Then, we need to eliminate the intermediate states one by one. Let us start with eliminating the sink state $e$. As a result, for all other states $i$ and $j$, we get $R'_{i,j} = R_{i,j} \cup (R_{i,e}^* R_{e,e} R_{e,j})$. By definition of a sink state, it has no arrows going from it to any other states. Thus, we always have $R_{e,j} = \emptyset$. Concatenation with the empty set $R^* e, j$ is empty set, so we always have $R_{i,e} R_{e,e} R_{e,j} = \emptyset$. Union of any set with the empty set is that same original set, so we have $R'_{i,j} = R_{i,j}$. Thus, we get the following simplified automaton.

Eliminating the start state. First, we draw all possible arrows:
Now, to find expressions to place at all these arrows, we will use the general formula

$$R'_{i,j} = R_{i,j} \cup (R_{i,k}^* R_{k,j}^*),$$

where $k$ is the state that we are eliminating, i.e., in this case, the state $k = s$.

By applying this formula, and by using simplification formulas described in the lecture, we get the following results:

$$R'_{ns,n} = R_{ns,n} \cup (R_{ns,s}^* R_{s,n}) = \emptyset \cup (\Lambda^* A) = \emptyset \cup (\Lambda^* \Lambda) = \emptyset \cup A = A;$$

$$R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,s}^* R_{s,nf}) = \emptyset \cup (\Lambda^* \emptyset) = \emptyset \cup \emptyset = \emptyset;$$

$$R'_{n,n} = R_{n,n} \cup (R_{n,s}^* R_{s,n}) = a \cup (\emptyset \ldots) = a \cup \emptyset = a;$$

$$R'_{n,nf} = R_{n,nf} \cup (R_{n,s}^* R_{s,nf}) = \Lambda \cup (\emptyset \ldots) = \Lambda \cup \emptyset = \Lambda.$$

Thus, the 3-state a-automaton takes the following form:

```
    n
   / \  \\
A   \   \  \\
 nS  \  \ nf
```

**Eliminating the state** $n$. Now, all that remains to do is to go from here to the 2-state a-automaton by eliminating the remaining state $n$:

```
    nS
   /  \  \\
A   \   \  \\
nS  \  \ nf
```

The final expression is the corresponding expression for $R'_{ns,nf}$:

$$R'_{ns,nf} = R_{ns,nf} \cup (R_{ns,s}^* R_{n,nf}) =$$

$$\emptyset \cup (Aa^* \Lambda) = Aa^* \Lambda = Aa^*.$$

**Resulting answer:** The regular expression corresponding to the original automaton is $Aa^*$. 