```
<program> ⇒ <function-list>
<function-list> ⇒ <function-list> <function>
<function-list> ⇒ <function>
<function> ⇒ int <identifier>(<formal-parameter-list>);
<function> ⇒ int <identifier>(<formal-parameter-list>) <compound-statement>
<formal-parameter-list> ⇒ <parameter-list>
<formal-parameter-list> ⇒ ε
<parammeter-list> ⇒ <parameter-list>,<parameter>
<parammeter-list> ⇒ <parameter>
<parameter> ⇒ int <parameter-declarator>
<parameter-declarator> ⇒ <identifier>
<parameter-declarator> ⇒ <parameter-declarator>[<number>]
<compound-statement> ⇒ {<declaration-list><statement-list>}
<declaration-list> ⇒ <declaration-list><variable-declaration>
<declaration-list> ⇒ ε
<variable-declaration> ⇒ int <id-list>;
<id-list> ⇒ <id-list>,<parameter-declarator>
<id-list> ⇒ <parameter-declarator>
<statement-list> ⇒ <statement-list><statement>
<statement-list> ⇒ ε
<statement> ⇒ <compound-statement>
<statement> ⇒ <assign-expression>;
<statement> ⇒ if (<assign-expression>) <statement> else <statement>
<statement> ⇒ while (<assign-expression>) <statement>
<statement> ⇒ return <assign-expression>;
<statement> ⇒ read <variable>;
<statement> ⇒ write <assign-expression>;
<statement> ⇒ writeln;
<assign-expression> ⇒ <variable> = <assign-expression>
<assign-expression> ⇒ <logical-expression>
<logical-expression> ⇒ !<relational-expression>
<logical-exprettion> ⇒ <relational-expression>
<relational-expression> ⇒ <expression> == <expression>
<relational-expression> ⇒ <expression> != <expression>
<relational-expression> ⇒ <expression> > <expression>
<relational-expression> ⇒ <expression> < <expression>
<relational-expression> ⇒ <expression> >= <expression>
<relational-expression> ⇒ <expression> <= <expression>
<relational-expression> ⇒ <expression>
<expression> ⇒ <expression> + <term>
<expression> ⇒ <expression> - <term>
<expression> ⇒ <term>
<term> ⇒ <term> * <factor>
<term> ⇒ <term> / <factor>
<term> ⇒ <factor>
<factor> ⇒ <identifier>(<expression-list>)
<factor> ⇒ (<assign-expression>)
<factor> ⇒ <variable>
<factor> ⇒ <number>
<variable> ⇒ <identifier>
<variable> ⇒ <variable>[<assign-expression>]
<expression-list> ⇒ <argument-list>
<expression-lst> ⇒ ε
<argument-list> ⇒ <argument-list>,<assign-expression>>
<argument-list> ⇒ <asssign-expression>
```