program	::=	(functiondefinition)* <EOF>
functiondefinition	::=	type <ID> "(" (<ID> ("," <ID>)*)? ")" ("{" |<KW_BEGIN>) statementlist ("}" |<KW_END>)
functioncall	::=	<ID> "(" (assignment ("," assignment)*)? ")"
statementlist	::=	(block)*
block	::=	("{" |<KW_BEGIN>) statementlist ("}" |<KW_END>)
	|	statement
statement	::=	ifstatement
	|	returnstatement ";"
	|	printf ";"
	|	statassignment ";"
	|	functioncall ";"
ifstatement	::=	<KW_IF> "(" assignment ")" block
returnstatement	::=	<KW_RETURN> (assignment)?
printf	::=	<KW_PRINTF> "(" assignment ")"
type	::=	<KW_BOOLEAN>
	|	<KW_FLOAT>
	|	<KW_INT>
	|	<KW_VOID>
statassignment	::=	<ID> "=" assignment
assignment	::=	((<ID> "=" assignment) |expr)
expr	::=	simpexpr (("==" |"!=" |"<=" |">=" |"<" |">") simpexpr)?
simpexpr	::=	("-")? term (("+" |"-" |"||") term)*
term	::=	factor (("*" |"/" |"&&") factor)*
factor	::=	<CONST_INT>
	|	<CONST_FLOAT>
	|	<CONST_BOOLEAN>
	|	functioncall
	|	<ID>
	|	"(" assignment ")"