The MyPL Syntax Rules

<program>	::=	(<struct_def> | <fun_def>)*
<struct_def>	::=	STRUCT ID LBRACE <fields> RBRACE
<fields>	::=	<data_type> ID (COMMA <data_type> ID)* | ϵ
<fun_def>	::=	(<data_type> | VOID_TYPE) ID LPAREN <params> RPAREN LBRACE (<stmt>)* RBRACE
<params>	::=	<data_type> ID (COMMA <data_type> ID)* | ϵ
<data_type>	::=	<base_type> | ID | ARRAY (<base_type> | ID)
<base_type>	::=	INT_TYPE | DOUBLE_TYPE | BOOL_TYPE | CHAR_TYPE | STRING_TYPE
<stmt>	::=	<vdecl_stmt> | <assign_stmt> | <if_stmt> | <while_stmt> | <for_stmt> | <call_expr> | <ret_stmt>
<vdecl_stmt>	::=	<data_type> ID ASSIGN <expr>
<assign_stmt>	::=	<lvalue> ASSIGN <expr>
<lvalue>	::=	ID (DOT ID | LBRACKET <expr> RBRACKET)*
<if_stmt>	::=	IF LPAREN <expr> RPAREN LBRACE (<stmt>)* RBRACE <if_stmt_t>
<if_stmt_t>	::=	ELSEIF LPAREN <expr> RPAREN LBRACE (<stmt>)* RBRACE <if_stmt_t> | ELSE LBRACE (<stmt>)* RBRACE | ϵ
<while_stmt>	::=	WHILE LPAREN <expr> RPAREN LBRACE (<stmt>)* RBRACE
<for_stmt>	::=	FOR LPAREN <vdecl_stmt> SEMICOLON <expr> SEMICOLON <assign_stmt> <RPAREN> LBRACE (<stmt>)* RBRACE
<call_expr>	::=	ID LPAREN (<expr> (COMMA <expr>)* | ϵ) RPAREN
<ret_stmt>	::=	RETURN <expr>
<expr>	::=	(<rvalue> | NOT <expr> | LPAREN <expr> RPAREN) (<bin_op> <expr> | ϵ)
<bin_op>	::=	PLUS | MINUS | TIMES | DIVIDE | AND | OR | EQUAL | LESS | GREATER | LESS_EQ | GREATER_EQ | NOT_EQUAL
<rvalue>	::=	<base_rvalue> | NULL_VAL | <new_rvalue> | <var_rvalue> | <call_expr>
<new_rvalue>	::=	NEW ID (LBRACKET <expr> RBRACKET | ϵ) | NEW <base_type> LBRACKET <expr> RBRACKET
<base_rvalue>	::=	INT_VAL | DOUBLE_VAL | BOOL_VAL | CHAR_VAL | STRING_VAL
<var_rvalue>	::=	ID (DOT ID | LBRACKET <expr> RBRACKET)*