%{

/\*

\* ========================================================================

\*

\* parse.y - Bison parser for Slic.

\* Added code to add variables to symbol table defined in stable.h.

\* Programmer --- Patrick Fischer

\*

\* ========================================================================

\*/

#include "stable.h"

#include <stdio.h>

#define DEBUG 0

struct symbol\_table \*table;

%}

%union {

char \*sval;

int ival;

struct symbol\_table \*tableptr;

struct symbol\_table\_entry \*entry;

}

%token RWMAIN

%token RWEXIT

%token RWIF

%token RWELSE

%token RWWHILE

%token RWCOUNTING

%token RWEND

%token RWDATA

%token RWALG

%token <ival> LITINT

%token LITREAL

%token RWINT

%token RWREAL

%token RWUPWARD

%token RWDOWNWARD

%token RWTO

%token RWREAD

%token RWPRINT

%token CHARSTRING

%token CARRETURN

%token ASSIGNOP

%token COMMA

%token SEMICOLON

%token COLON

%token LPAREN

%token RPAREN

%token LBRACK

%token RBRACK

%token ADD

%token MINUS

%token MULT

%token DIV

%token LESS

%token GREATER

%token LESSEQU

%token GREATEQU

%token EQUAL

%token NOTEQUAL

%token AND

%token OR

%token NOT

%token <sval> VAR

%token <sval> VARIABLE

%type <entry> decstmt

%type <entry> varlist

%type <entry> varref

%%

program : headingstmt datasection algsection endmainstmt;

headingstmt: RWMAIN SEMICOLON;

datasection: RWDATA COLON

|RWDATA COLON decstmtlist

; // decstmtlist is a listing of the variables used by this Slic program.

decstmtlist: decstmtlist decstmt

|decstmt

; // decstmt is just a variable declaration.

decstmt: RWINT COLON varlist {

$$ = malloc(sizeof(struct symbol\_table\_entry));

$$->type=TYPE\_INT;

$$->name=$3->name;

$$->kind = $3->kind;

$$->address = $3->address;

$$->size = $3->size;

insert(\*$$);

}

|RWREAL COLON varlist { $$ = malloc(sizeof(struct symbol\_table\_entry));

$$->type=TYPE\_REAL;

$$->name=$3->name;

$$->kind = $3->kind;

$$->address = $3->address;

$$->size = $3->size;

insert(\*$$);

}

; // The reserved word is followed by a list because any number of variables can be declared on a single line.

varlist: varref COMMA varlist

|varref SEMICOLON

;

varref: VAR { $$ = malloc(sizeof(struct symbol\_table\_entry));

$$->name = $1;

if(DEBUG) printf("$1 = %s\n", $1);

$$->address = 0;

$$->kind = KIND\_SCALAR;

$$->size = 1;

}

|VAR LBRACK LITINT RBRACK {

$$ = malloc(sizeof(struct symbol\_table\_entry));

$$->name = $1;

$$->address = 0;

$$->kind = KIND\_ARRAY;

$$->size = $3;

}

; // varref refers to the VAR token or an array.

algsection: RWALG COLON;

endmainstmt: RWEND RWMAIN SEMICOLON;

%%

int yyerror() {

printf("Called yyerror()\n");

return 0;

}