Moldovan Vasilica – Lab9 FLCD

Yacc/Bison

%{  
#include <stdio.h>  
#include <stdlib.h>  
  
int yylex();  
int yyerror(char \*s);  
extern FILE \*yyin;  
extern int yylineno;  
%}  
  
%token IDENTIFIER  
%token CONSTANT  
%token WORD  
%token INDIVIDUAL  
%token DECISION  
%token CHAR  
%token FLOAT  
%token CONST  
%token PARSING  
%token SITUATION  
%token OTHER  
%token COME  
%token LEAVE  
%token RETURN  
%token BREAK  
%token COLON  
%token SEMI\_COLON  
%token COMA  
%token DOT  
%token PLUS  
%token MINUS  
%token MULTIPLY  
%token DIVISION  
%token LEFT\_ROUND\_PARENTHESIS  
%token RIGHT\_ROUND\_PARENTHESIS  
%token LEFT\_SQUARE\_PARENTHESIS  
%token RIGHT\_SQUARE\_PARENTHESIS  
%token LESS\_THAN  
%token GREATER\_THAN  
%token LESS\_OR\_EQUAL\_THAN  
%token GREATER\_OR\_EQUAL\_THAN  
%token DIFFERENT  
%token EQUAL  
%token ASSIGNMENT  
%token OR  
%token AND  
%token LEFT\_CURLY\_BRACKET  
%token RIGHT\_CURLY\_BRACKET  
  
%start program  
  
%%  
  
program : declist cmpdstmt  
declist : declaration | declaration declist  
declaration: type identifierList SEMI\_COLON | arraydecl SEMI\_COLON  
identifierList : IDENTIFIER | IDENTIFIER COMA identifierList  
type : INDIVIDUAL | DECISION | CHAR | FLOAT  
arraydecl : type IDENTIFIER LEFT\_SQUARE\_PARENTHESIS CONSTANT RIGHT\_SQUARE\_PARENTHESIS  
cmpdstmt : stmt | stmt cmpdstmt  
stmt : simplstmt SEMI\_COLON | structstmt  
simplstmt : assignstmt | iostmt  
assignstmt : IDENTIFIER ASSIGNMENT expression  
iostmt : COME IDENTIFIER | LEAVE IDENTIFIER | COME CONSTANT | LEAVE CONSTANT | COME WORD | LEAVE WORD  
expression : expression PLUS term | expression MINUS term | expression MULTIPLY term | expression DIVISION term | term  
term : IDENTIFIER | CONSTANT | arrElem  
arrElem : IDENTIFIER LEFT\_SQUARE\_PARENTHESIS CONST RIGHT\_SQUARE\_PARENTHESIS | IDENTIFIER LEFT\_SQUARE\_PARENTHESIS IDENTIFIER RIGHT\_SQUARE\_PARENTHESIS  
structstmt : whilestmt | ifstmt  
ifstmt : SITUATION LEFT\_ROUND\_PARENTHESIS condition RIGHT\_ROUND\_PARENTHESIS LEFT\_CURLY\_BRACKET stmt RIGHT\_CURLY\_BRACKET OTHER LEFT\_CURLY\_BRACKET stmt RIGHT\_CURLY\_BRACKET | SITUATION LEFT\_ROUND\_PARENTHESIS condition RIGHT\_ROUND\_PARENTHESIS LEFT\_CURLY\_BRACKET stmt RIGHT\_CURLY\_BRACKET

whilestmt : PARSING LEFT\_ROUND\_PARENTHESIS condition RIGHT\_ROUND\_PARENTHESIS LEFT\_CURLY\_BRACKET stmt RIGHT\_CURLY\_BRACKET

%%  
int yyerror(char \*s)  
{  
 printf("%s on line %d\n", s, yylineno);  
 return 0;  
}  
  
int main(int argc, char\*\* argv)  
{  
 if (argc == 2) {  
 yyin = fopen(argv[1], "r");  
 yyparse();  
 }  
 yyparse();  
 return 0;  
}

p1:

Reserved word: individual

Identifier: a

Separator: ,

Identifier: b

Separator: ,

Identifier: gcd

Separator: ;

Reserved word: come

Identifier: a

Separator: ;

Reserved word: come

Identifier: b

Separator: ;

Reserved word: parsing

Separator: (

Identifier: a

Operator: !=

Identifier: b

Separator: )

Reserved word: situation

Separator: (

Identifier: a

Operator: >

Identifier: b

Separator: )

Identifier: a

Operator: =

Identifier: a

Operator: -

Identifier: b

Separator: ;

Reserved word: other

Identifier: b

Operator: =

Identifier: b

Operator: -

Identifier: a

Separator: ;

Identifier: gcd

Operator: =

Identifier: a

Separator: ;

Reserved word: leave

Identifier: gcd

Separator: ;

Done

p2:

Reserved word: individual

Identifier: arr

Separator: [

Constant: 100

Separator: ]

Separator: ;

Reserved word: decision

Identifier: isSmaller

Separator: ;

Reserved word: individual

Identifier: n

Separator: ,

Identifier: i

Separator: ;

Reserved word: individual

Identifier: maxNumber

Separator: ;

Identifier: isSmaller

Operator: =

Constant: 1

Separator: ;

Reserved word: come

Identifier: n

Separator: ;

Reserved word: come

Identifier: m

Separator: ;

Identifier: i

Operator: =

Constant: 0

Separator: ;

Reserved word: parsing

Separator: (

Identifier: i

Operator: <

Identifier: n

Separator: )

Reserved word: come

Identifier: arr

Separator: [

Identifier: i

Separator: ]

Separator: ;

Reserved word: situation

Separator: (

Identifier: i

Operator: ==

Constant: 0

Separator: )

Identifier: maxNumber

Operator: =

Identifier: arr

Separator: [

Identifier: i

Separator: ]

Separator: ;

Reserved word: other

Reserved word: situation

Separator: (

Identifier: maxNumber

Operator: <

Identifier: arr

Separator: [

Identifier: i

Separator: ]

Separator: )

Identifier: maxNumber

Operator: =

Identifier: arr

Separator: [

Identifier: i

Separator: ]

Separator: ;

Identifier: i

Operator: =

Identifier: i

Operator: +

Constant: 1

Separator: ;

Reserved word: situation

Separator: (

Identifier: maxNumber

Operator: >=

Identifier: m

Separator: )

Identifier: isSmaller

Operator: =

Constant: 0

Separator: ;

Reserved word: leave

Word "The maximum number is "

Separator: ;

Reserved word: leave

Identifier: maxNumber

Separator: ;

Reserved word: leave

Word " and is smaller than "

Separator: ;

Reserved word: leave

Identifier: m

Separator: ;

Reserved word: leave

Word " - "

Separator: ;

Reserved word: leave

Identifier: isSmaller

Separator: ;

Done

p3:

Reserved word: individual

Identifier: arr

Separator: [

Constant: 100

Separator: ]

Separator: ;

Reserved word: decision

Identifier: isSmaller

Separator: ;

Reserved word: individual

Identifier: n

Separator: ,

Identifier: i

Separator: ;

Reserved word: individual

Identifier: maxNumber

Separator: ;

Identifier: isSmaller

Operator: =

Constant: 1

Separator: ;

Reserved word: come

Identifier: n

Separator: ;

Reserved word: come

Identifier: m

Separator: ;

Identifier: i

Operator: =

Constant: 0

Separator: ;

Reserved word: parsing

Separator: (

Identifier: i

Operator: <

Identifier: n

Separator: )

Reserved word: come

Identifier: arr

Separator: [

Identifier: i

Separator: ]

Separator: ;

Separator: ;

syntax error on line 10