

## 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