<https://github.com/pauladam2001/Sem5_FormalLanguagesAndCompilerDesign/tree/master/Lab9>

**Yacc Specification File**

%{

#include <stdio.h>

#include <stdlib.h>

#define YYDEBUG 1

%}

%token INTEGER

%token STRING

%token CHAR

%token WHILE

%token FOR

%token IF

%token ELSEIF

%token ELSE

%token READ

%token PUTS

%token BREAK

%token RETURN

%token NEXT

%token END

%token plus

%token minus

%token mul

%token division

%token eq

%token equal

%token different

%token less

%token more

%token lessOrEqual

%token moreOrEqual

%token leftRoundBracket

%token rightRoundBracket

%token leftCurlyBracket

%token rightCurlyBracket

%token IDENTIFIER

%token NUMBER\_CONST

%token STRING\_CONST

%token CHAR\_CONST

%start program

%%

program : declaration\_list statements

declaration\_list : declaration declaration\_list | /\*Empty\*/

declaration : var\_type IDENTIFIER equal\_expression

equal\_expression : eq expression | /\*Empty\*/

var\_type : INTEGER | CHAR | STRING

expression : term sign\_and\_expression

sign\_and\_expression : sign expression | /\*Empty\*/

sign : plus | minus | mul | division

term : IDENTIFIER | constant

constant : NUMBER\_CONST | STRING\_CONST | CHAR\_CONST

statements : statement statements | /\*Empty\*/

statement : simple\_stmt | struct\_stmt

simple\_stmt : assignment\_stmt | input\_output\_stmt

struct\_stmt : if\_stmt | while\_stmt

assignment\_stmt : IDENTIFIER eq expression

input\_output\_stmt : READ leftRoundBracket term rightRoundBracket | PUTS leftRoundBracket term rightRoundBracket

if\_stmt : IF leftRoundBracket condition rightRoundBracket leftCurlyBracket statements rightCurlyBracket else\_stmt

else\_stmt : ELSE leftCurlyBracket statements rightCurlyBracket | /\*Empty\*/

while\_stmt : WHILE leftRoundBracket condition rightRoundBracket leftCurlyBracket statements rightCurlyBracket

condition : expression relation expression

relation : equal | different | less | more | lessOrEqual | moreOrEqual

%%

yyerror(char \*s)

{

printf("%s\n",s);

}

extern FILE \*yyin;

main(int argc, char \*\*argv)

{

if(argc>1) yyin : fopen(argv[1],"r");

if(argc>2 && !strcmp(argv[2],"-d")) yydebug: 1;

if(!yyparse()) fprintf(stderr, "\tProgram is syntactically correct.\n");

}

**Demo**

Run:

