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

Yacc Specification File

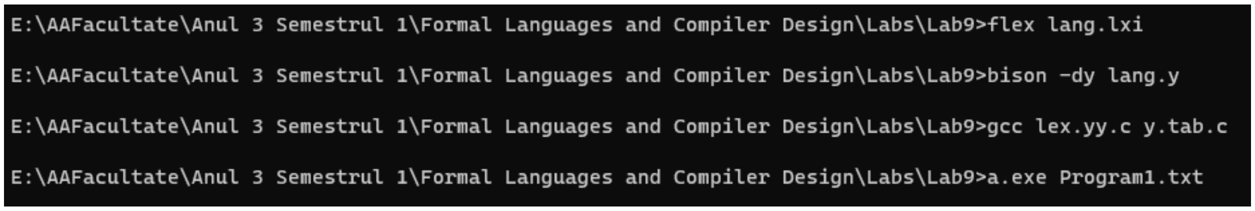
```
%{
#include <stdio.h>
#include <stdlib.h>

#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:



```
E:\AAFacultate\Anul 3 Semestrul 1\Formal Languages and Compiler Design\Labs\Lab9>flex lang.lxi

E:\AAFacultate\Anul 3 Semestrul 1\Formal Languages and Compiler Design\Labs\Lab9>bison -dy lang.y

E:\AAFacultate\Anul 3 Semestrul 1\Formal Languages and Compiler Design\Labs\Lab9>gcc lex.yy.c y.tab.c

E:\AAFacultate\Anul 3 Semestrul 1\Formal Languages and Compiler Design\Labs\Lab9>a.exe Program1.txt
```