EXP NO -7

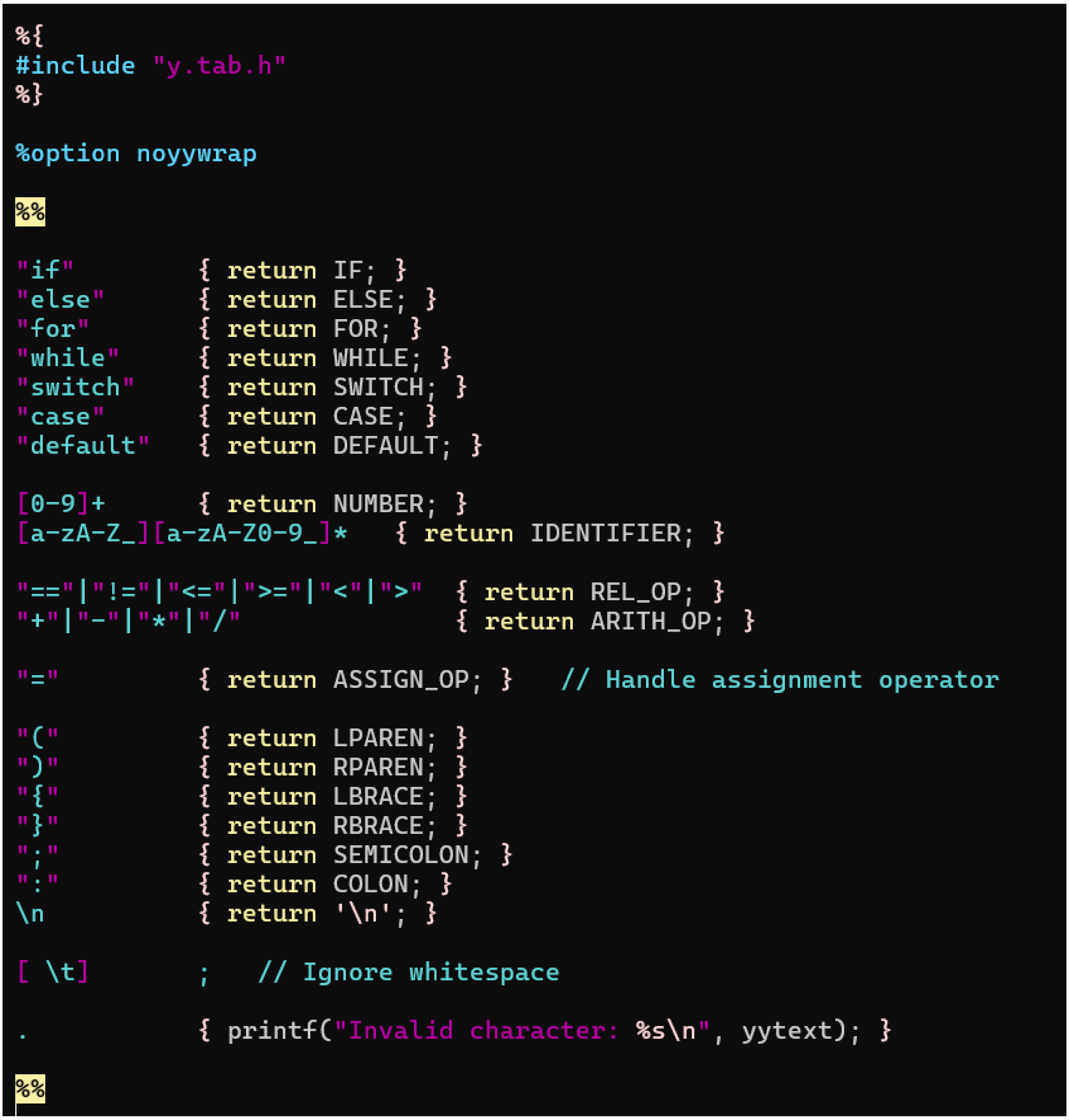
RECOGNIZE A VALID CONTROL STRUCTURES SYNTAX OF C LANGUAGE (FOR LOOP, WHILE LOOP, IF-ELSE, IF-ELSE-IF, SWITCH CASE, ETC.,)

AIM:

To design and implement a LEX and YACC program that recognizes the syntax of common control structures in C programming.

PROGRAM

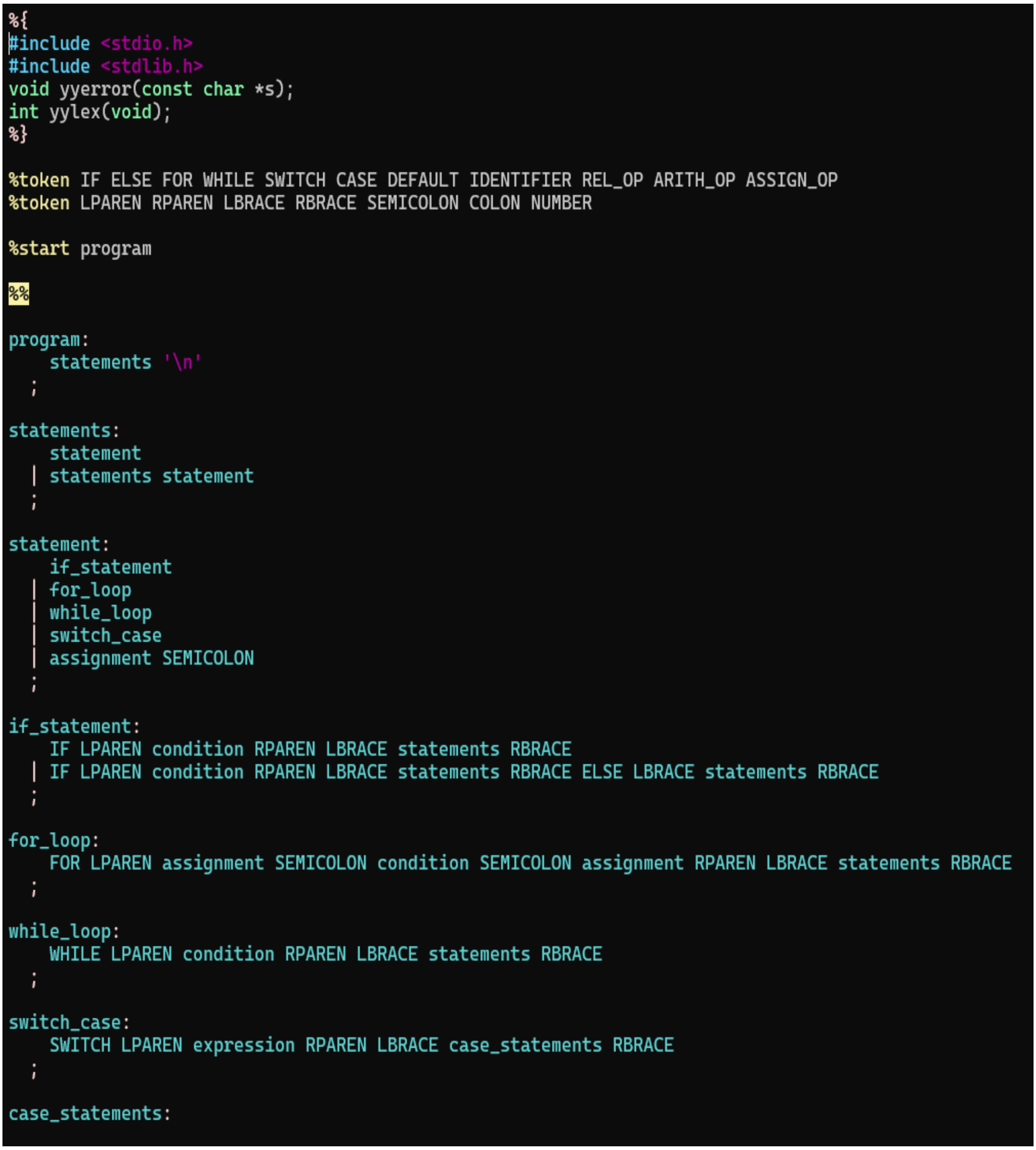
LEX CODE: cs.l



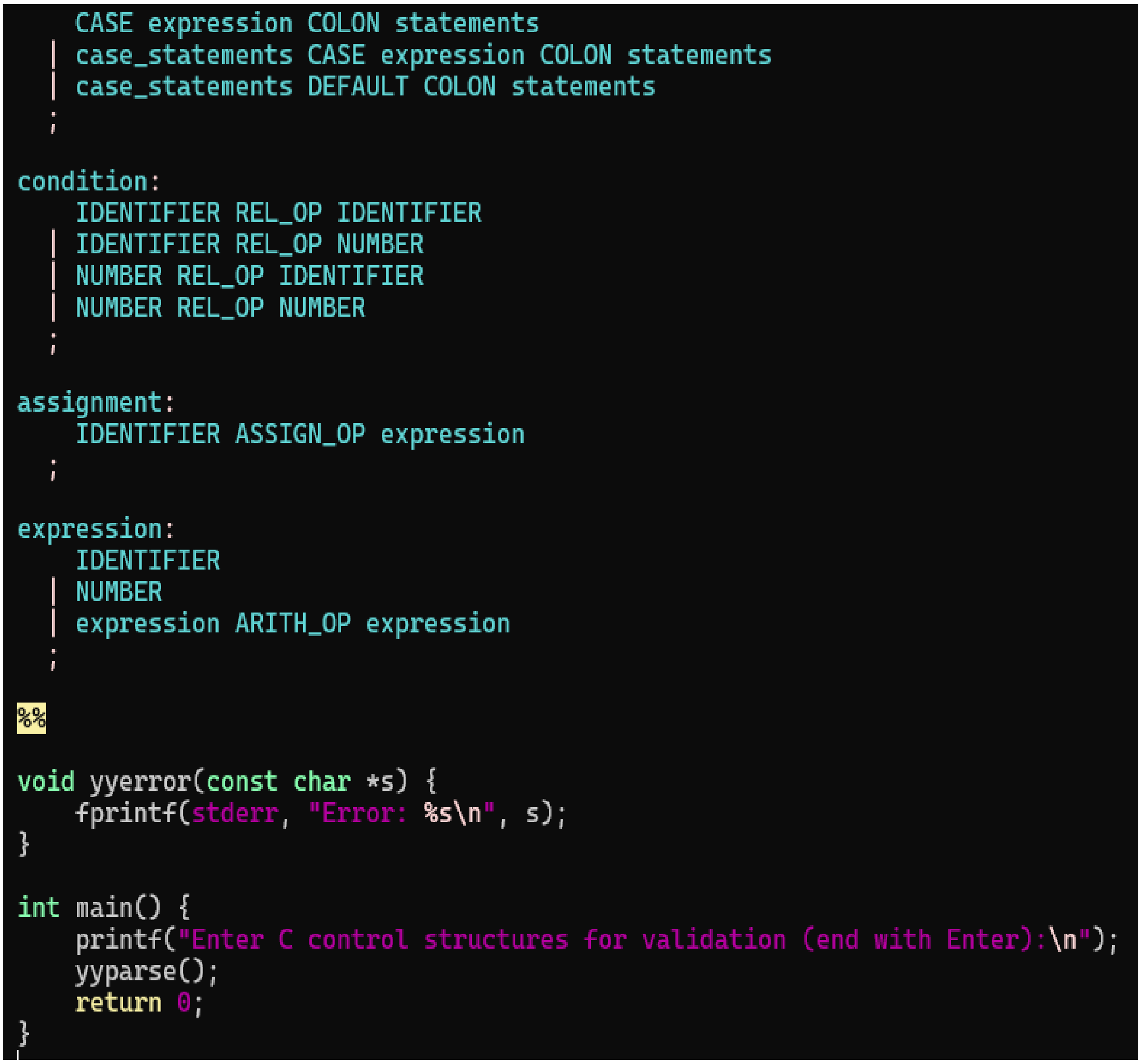
```
%{
#include "y.tab.h"
%}

%option noyywrap

%%

"if"        { return IF; }
"else"      { return ELSE; }
"for"       { return FOR; }
"while"     { return WHILE; }
"switch"    { return SWITCH; }
"case"      { return CASE; }
"default"   { return DEFAULT; }

[0-9]+      { return NUMBER; }
[a-zA-Z_][a-zA-Z0-9_]*   { return IDENTIFIER; }

"=="|"!="|"<="|">="|"<"|">"  { return REL_OP; }
"+"|"-"|"*"|"/"              { return ARITH_OP; }

"="         { return ASSIGN_OP; }   // Handle assignment operator

"("         { return LPAREN; }
")"         { return RPAREN; }
"{"         { return LBRACE; }
"}"         { return RBRACE; }
";"         { return SEMICOLON; }
":"         { return COLON; }
\n          { return '\n'; }

[ \t]       ;   // Ignore whitespace

.           { printf("Invalid character: %s\n", yytext); }

%%
```

KAMALI K A – 220701118

YACC CODE: cs.y

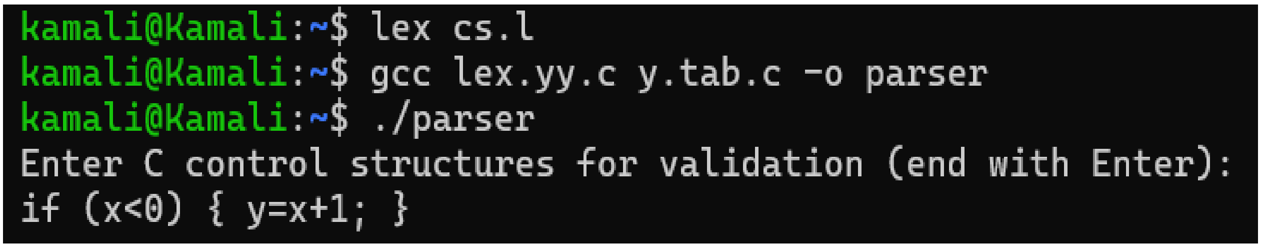


```
%{
#include <stdio.h>
#include <stdlib.h>
void yyerror(const char *s);
int yylex(void);
%}

%token IF ELSE FOR WHILE SWITCH CASE DEFAULT IDENTIFIER REL_OP ARITH_OP ASSIGN_OP
%token LPAREN RPAREN LBRACE RBRACE SEMICOLON COLON NUMBER

%start program

%%

program:
    statements '\n'
  ;

statements:
    statement
  | statements statement
  ;

statement:
    if_statement
  | for_loop
  | while_loop
  | switch_case
  | assignment SEMICOLON
  ;

if_statement:
    IF LPAREN condition RPAREN LBRACE statements RBRACE
  | IF LPAREN condition RPAREN LBRACE statements RBRACE ELSE LBRACE statements RBRACE
  ;

for_loop:
    FOR LPAREN assignment SEMICOLON condition SEMICOLON assignment RPAREN LBRACE statements RBRACE
  ;

while_loop:
    WHILE LPAREN condition RPAREN LBRACE statements RBRACE
  ;

switch_case:
    SWITCH LPAREN expression RPAREN LBRACE case_statements RBRACE
  ;

case_statements:
```



```
    CASE expression COLON statements
  | case_statements CASE expression COLON statements
  | case_statements DEFAULT COLON statements
  ;

condition:
    IDENTIFIER REL_OP IDENTIFIER
  | IDENTIFIER REL_OP NUMBER
  | NUMBER REL_OP IDENTIFIER
  | NUMBER REL_OP NUMBER
  ;

assignment:
    IDENTIFIER ASSIGN_OP expression
  ;

expression:
    IDENTIFIER
  | NUMBER
  | expression ARITH_OP expression
  ;

%%

void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}

int main() {
    printf("Enter C control structures for validation (end with Enter):\n");
    yyparse();
    return 0;
}
```

OUTPUT



```
kamali@Kamali:~$ lex cs.l
kamali@Kamali:~$ gcc lex.yy.c y.tab.c -o parser
kamali@Kamali:~$ ./parser
Enter C control structures for validation (end with Enter):
if (x<0) { y=x+1; }
```

RESULT:

Thus the above program to recognize a valid control structures syntax of c language (for loop, while loop, if-else, if-else-if, switch case as been implemented and executed successfully with LEX and YACC.

KAMALI K A – 220701118