AIR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Lab Task 7

Student Name: Hamza Umer Farooq **Reg. No: 200789**

Subject: Compiler Construction **Semester: VIII**

Objective: Identify IF, WHILE, FOR

ASSESSMENT:

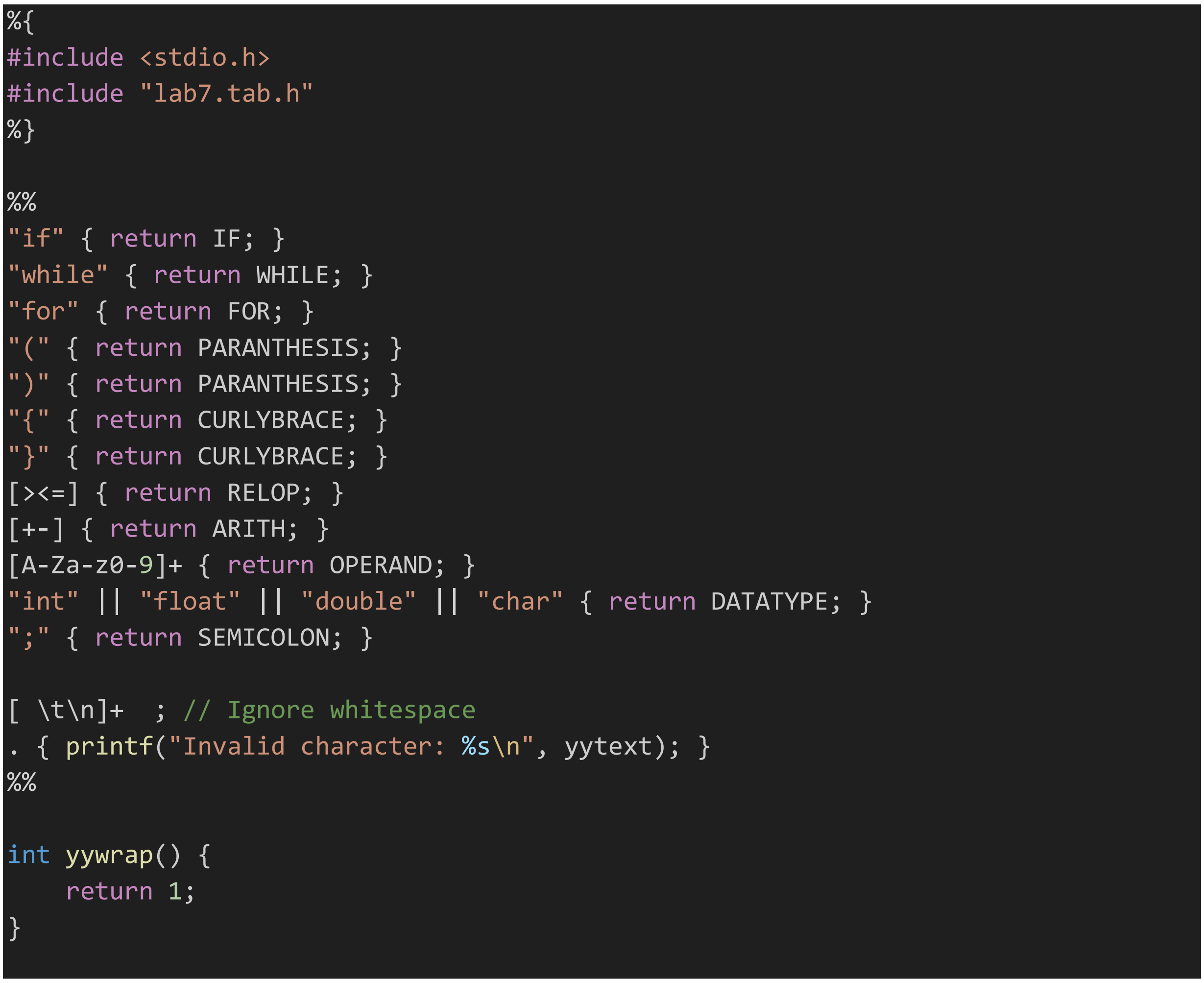
Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Ability to Conduct Task					
Ability to assimilate the results					
Effective use of theorems/postulates/formulas					

Total Marks:
Obtained Marks:

REPORT ASSESSMENT:

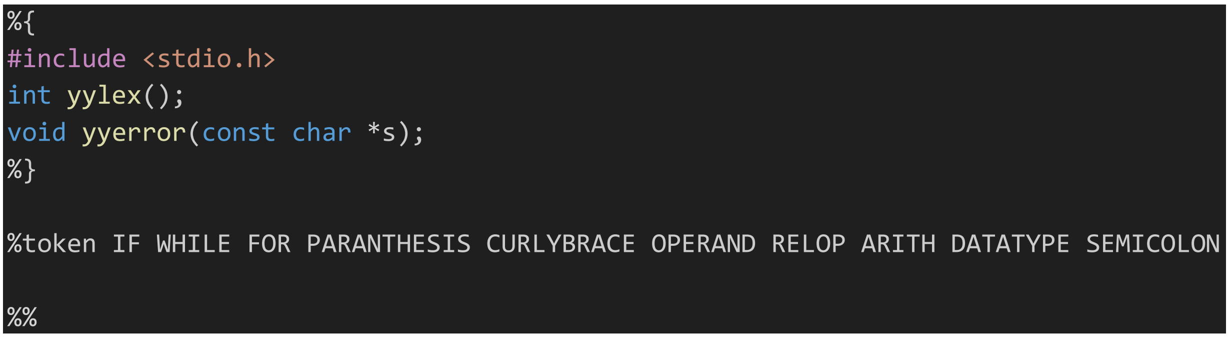
Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Data presentation					
Experimental results					
Conclusion					

Lab7.l:

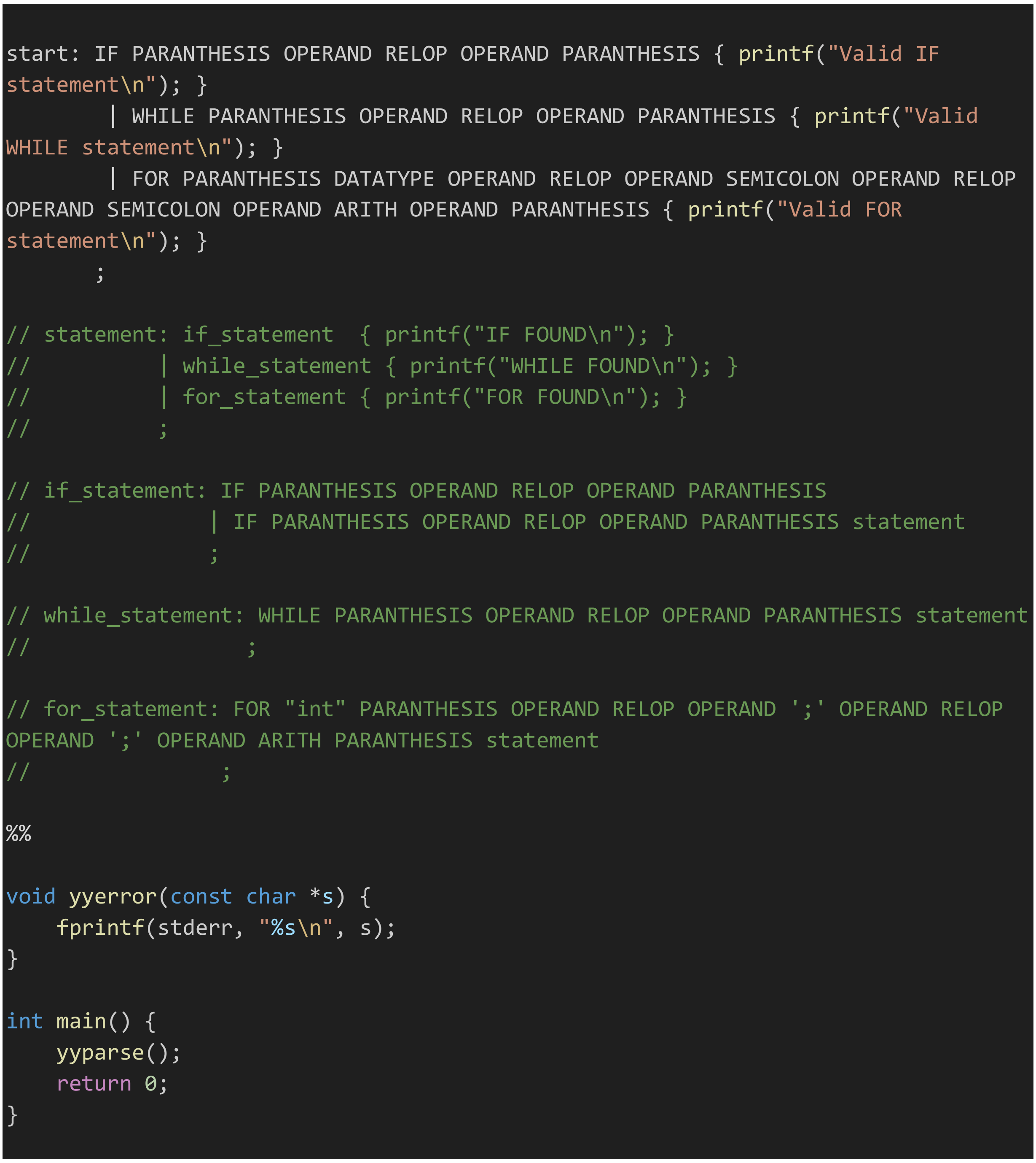


```
%{
#include <stdio.h>
#include "lab7.tab.h"
%}

%%
"if" { return IF; }
"while" { return WHILE; }
"for" { return FOR; }
"(" { return PARANTHESIS; }
")" { return PARANTHESIS; }
"{" { return CURLYBRACE; }
"}" { return CURLYBRACE; }
[><=] { return RELOP; }
[+-] { return ARITH; }
[A-Za-z0-9]+ { return OPERAND; }
"int" || "float" || "double" || "char" { return DATATYPE; }
";" { return SEMICOLON; }

[ \t\n]+  ; // Ignore whitespace
. { printf("Invalid character: %s\n", yytext); }
%%

int yywrap() {
    return 1;
}
```

Lab7.y:

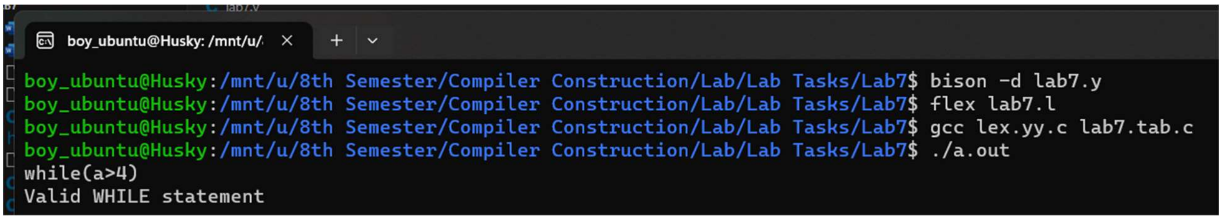


```
%{
#include <stdio.h>
int yylex();
void yyerror(const char *s);
%}

%token IF WHILE FOR PARANTHESIS CURLYBRACE OPERAND RELOP ARITH DATATYPE SEMICOLON

%%
```

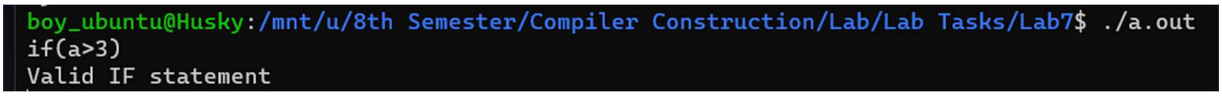


```
start: IF PARANTHESIS OPERAND RELOP OPERAND PARANTHESIS { printf("Valid IF statement\n"); }
        | WHILE PARANTHESIS OPERAND RELOP OPERAND PARANTHESIS { printf("Valid WHILE statement\n"); }
        | FOR PARANTHESIS DATATYPE OPERAND RELOP OPERAND SEMICOLON OPERAND RELOP OPERAND SEMICOLON OPERAND ARITH OPERAND PARANTHESIS { printf("Valid FOR statement\n"); }
       ;

// statement: if_statement  { printf("IF FOUND\n"); }
//          | while_statement { printf("WHILE FOUND\n"); }
//          | for_statement { printf("FOR FOUND\n"); }
//          ;

// if_statement: IF PARANTHESIS OPERAND RELOP OPERAND PARANTHESIS
//              | IF PARANTHESIS OPERAND RELOP OPERAND PARANTHESIS statement
//              ;

// while_statement: WHILE PARANTHESIS OPERAND RELOP OPERAND PARANTHESIS statement
//                 ;

// for_statement: FOR "int" PARANTHESIS OPERAND RELOP OPERAND ';' OPERAND RELOP OPERAND ';' OPERAND ARITH PARANTHESIS statement
//               ;

%%

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

int main() {
    yyparse();
    return 0;
}
```

OUTPUT



```
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab7$ bison -d lab7.y
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab7$ flex lab7.l
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab7$ gcc lex.yy.c lab7.tab.c
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab7$ ./a.out
while(a>4)
Valid WHILE statement
```



```
boy_ubuntu@Husky:/mnt/u/8th Semester/Compiler Construction/Lab/Lab Tasks/Lab7$ ./a.out
if(a>3)
Valid IF statement
```