1.Write a yacc program to implement relational operators

CODE :-

lex

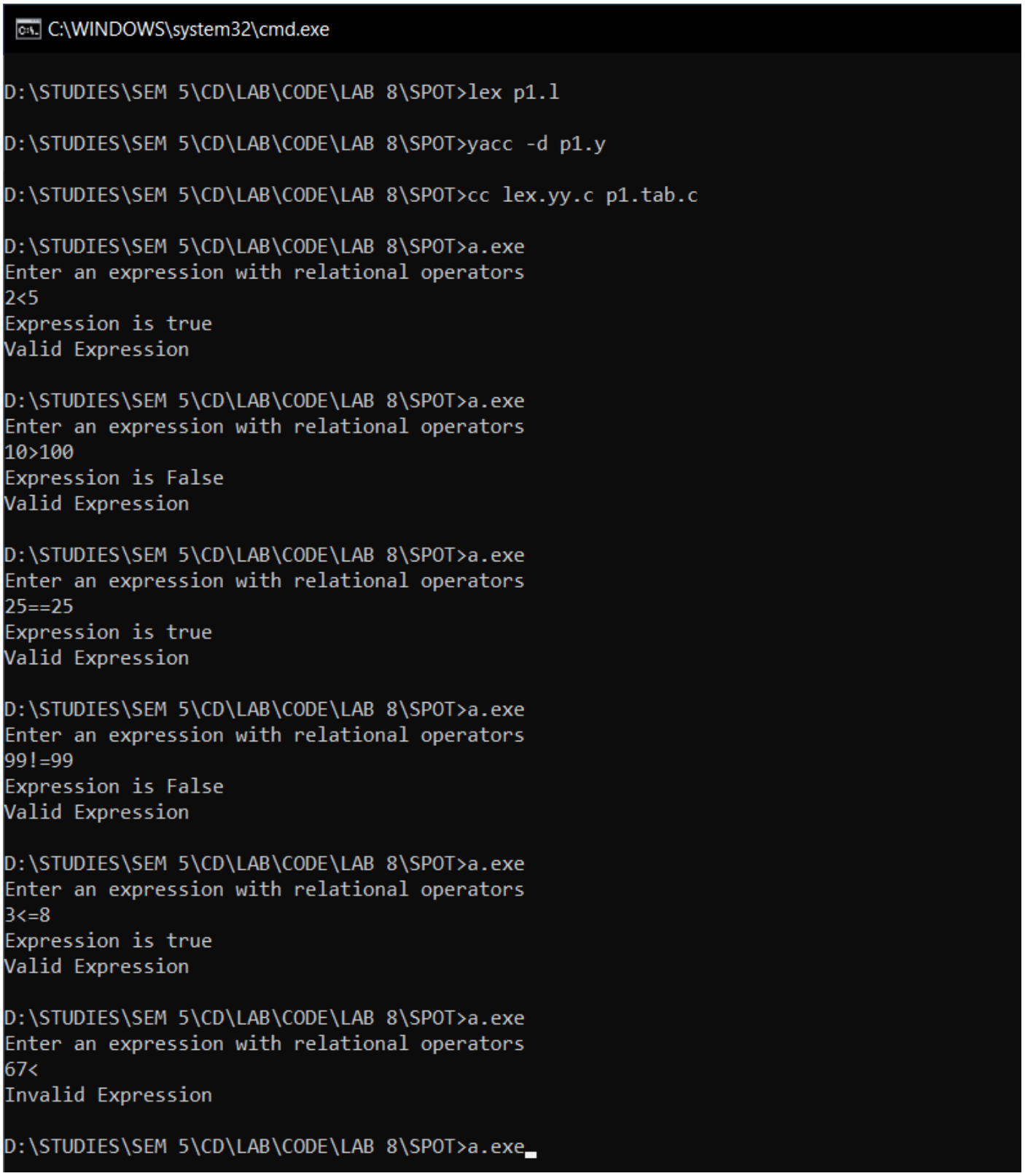
```
%{
    #include<stdio.h>
    #include "p1.tab.h"
    extern int yylval;
%}

%%
[0-9]+ {
        yylval=atoi(yytext);
        return NUMBER;
}
"==" {return EQ;}
">=" {return GE;}
"<=" {return LE;}
"!=" {return NE;}
[\t] ;
[\n] return 0;
. return yytext[0];
%%
int yywrap()
{
        return 1;
}
```

yacc

```
%{
        #include<stdio.h>
        int flag=0;
        int yylex();
        void yyerror();
%}
%token NUMBER LE GE EQ NE
%left '>' '<' LE GE EQ NE
%%
logicalExpression: E{
        if($$ == 1)
                printf("Expression is true\n");
        else
                printf("Expression is False\n");
        return 0;
};
E :E'>'E {$$=$1>$3;}
|E'<'E {$$=$1<$3;}
|E GE E {$$=$1>=$3;}
|E LE E {$$=$1<=$3;}
|E EQ E {$$=$1==$3;}
|E NE E {$$=$1!=$3;}
| NUMBER {$$=$1;}
;
%%
void main()
{
        printf("Enter an expression with relational operators\n");
        yyparse();
        if(flag==0)
                printf("Valid Expression\n");
}
void yyerror()
{
        printf("Invalid Expression\n");
        flag=1;
}
```

OUTPUT :-


C:\WINDOWS\system32\cmd.exe
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>lex p1.l
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>yacc -d p1.y
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>cc lex.yy.c p1.tab.c
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
2<5
Expression is true
Valid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
10>100
Expression is False
Valid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
25==25
Expression is true
Valid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
99!=99
Expression is False
Valid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
3<=8
Expression is true
Valid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe
Enter an expression with relational operators
67<
Invalid Expression
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe

2.Write a yacc program to verify the syntax of the given if-else statements

CODE :-

lex

```
%option noyywrap
%{
    #include <stdio.h>
    #include "p2.tab.h"
%}

alpha [a-zA-Z]
digit [0-9]

%%

[\t \n]

if  {   return IF;  }
else    {   return ELSE;    }
{digit}+    {   return NUM; }
{alpha}({alpha}|{digit})*   {   return ID;  }

"<="    {   return LE;  }
">="    {   return GE;  }
"=="    {   return EQ;  }
"!="    {   return NE;  }
"||"    {   return OR;  }
"&&"    {   return AND; }

.+; {   return LINE;    }
.   {   return yytext[0];   }

%%
```

yacc

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

%token ID NUM IF ELSE LE GE EQ NE OR AND LINE
%left OR AND
%left '>' '<' LE GE EQ NE
%left '!'

%%

S: ST   {
    printf("\nInput Accepted\n\n");
    return 0;
}

ST: IF '(' E ')' DEF ELSE DEF;

DEF:
    '{' BODY '}'
    | LINE
    | ST
    |
;

BODY:
    BODY BODY
    | LINE
    | ST
    |
;
```

```
35
36  E:
37       | E '<' E
38       | E '>' E
39       | E LE E
40       | E GE E
41       | E EQ E
42       | E NE E
43       | E OR E
44       | E AND E
45       | '(' E ')'
46       | ID
47       | NUM
48  ;
49
50  %%
51
52  int main() {
53      printf("\nEnter the expression:\n");
54      yyparse();
55      return 0;
56  }
57
58  void yyerror() {
59      printf("\nInvalid syntax\n\n");
60  }
```

OUTPUT :-

```
C:\WINDOWS\system32\cmd.exe
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>lex p2.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>yacc -d p2.y

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>cc lex.yy.c p2.tab.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe

Enter the expression:
if(x<y){
statements;
}
else{
statements;
}

Input Accepted


D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe

Enter the expression:
if(a==5)
        statements;
ellsee

Invalid syntax
```

```
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe

Enter the expression:
iff(x<=10)){

Invalid syntax


D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>a.exe

Enter the expression:
if(true){
        statements;
}
else{
        statements;
}

Input Accepted


D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 8\SPOT>
```