Practical 7

Name: Rajatkumar Patel
Roll No.: 18BCE191

Aim

To implement grammar rules for control statements and Loop control.

Code

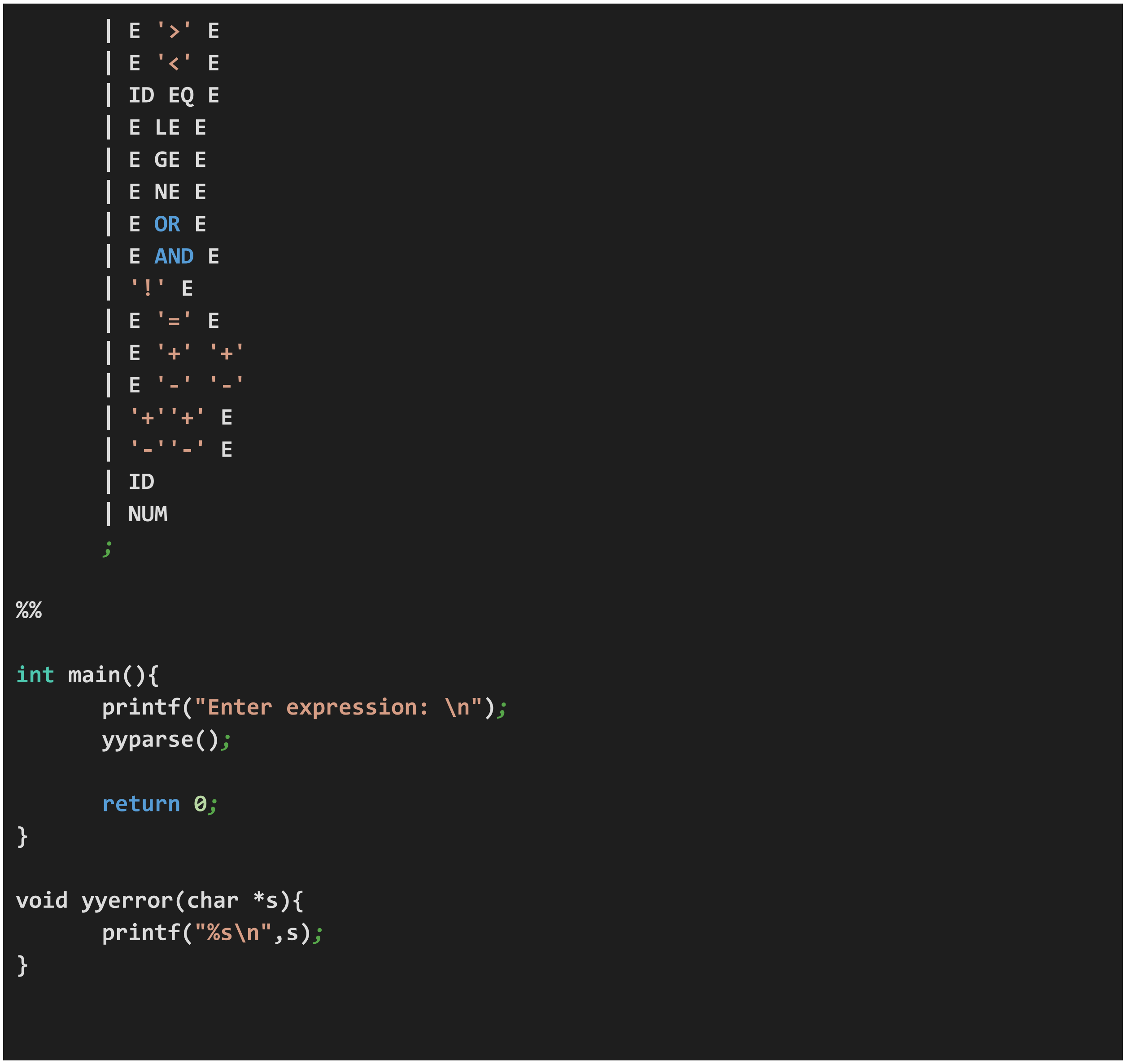
Input

Practical-7.l file

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

%%

"if" {return IF;}
"else" {return ELSE;}
"switch" {return SWITCH;}
"case" {return CASE;}
"default" {return DEFAULT;}
"for" {return FOR;}
"while" {return WHILE;}
[0-9]+ {return NUM;}
[A-Za-z][A-Za-z0-9]* {return ID;}
"<=" {return LE;}
">=" {return GE;}
"==" {return EQ;}
"!=" {return NE;}
"||" {return OR;}
"&&" {return AND;}
\n
. {return yytext[0];}

%%

yywrap(){
	return 1;
}
```

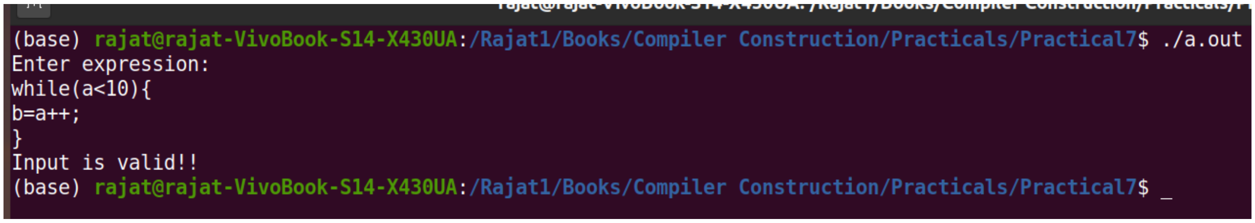
Practical-7.y file

```
%{

#include <stdio.h>
#include <stdlib.h>

%}

%token IF ELSE SWITCH CASE DEFAULT NUM ID FOR WHILE LE GE EQ NE OR AND

%right '='
%left OR AND
%left '>' '<' LE GE EQ NE
%left '+' '-'
%left '*' '/'
%left '!'

%%

S : ST { printf("Input is valid!! \n") ; exit(0);}
  ;

ST : IF '(' E ')' BLOCK
   | IF '(' E ')' BLOCK ELSE BLOCK
   | SWITCH '(' E ')' '{' CBLOCK '}'
   | FOR '(' E ';' E ';' E ')' BLOCK
   | WHILE '(' E ')' BLOCK
   ;

CBLOCK : CASE ' ' NUM':' BLOCK
         | DEFAULT ':' BLOCK
         ;

BLOCK: E ';'
       | '{' BODY '}'
       ;

BODY : E ';'
       | ST
       ;

E      : E '+' E
       | E '-' E
       | E    '*' E
       | E '/' E
```

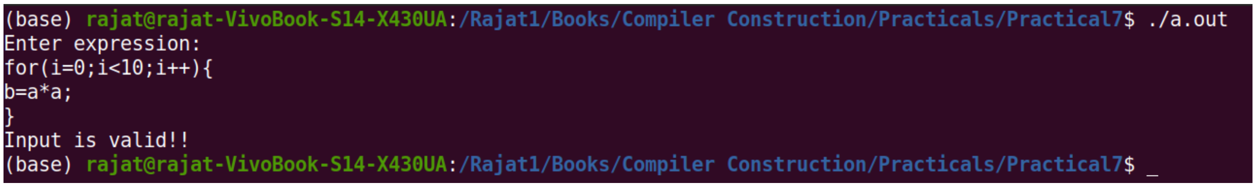


```
        | E '>' E
        | E '<' E
        | ID EQ E
        | E LE E
        | E GE E
        | E NE E
        | E OR E
        | E AND E
        | '!' E
        | E '=' E
        | E '+' '+'
        | E '-' '-'
        | '+''+' E
        | '-''-' E
        | ID
        | NUM
        ;

%%

int main(){
        printf("Enter expression: \n");
        yyparse();

        return 0;
}

void yyerror(char *s){
        printf("%s\n",s);
}
```

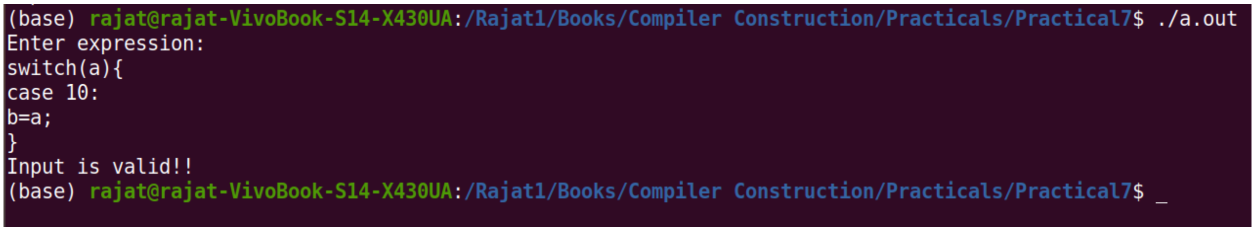
Output



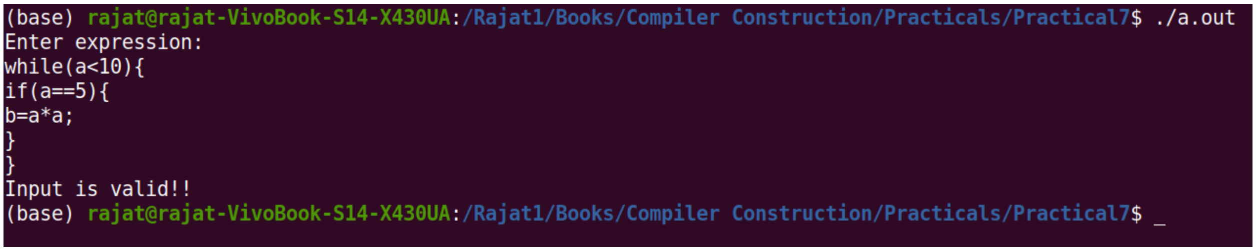
```
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ ./a.out
Enter expression:
while(a<10){
b=a++;
}
Input is valid!!
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ _
```



```
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ ./a.out
Enter expression:
for(i=0;i<10;i++){
b=a*a;
}
Input is valid!!
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ _
```



```
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ ./a.out
Enter expression:
switch(a){
case 10:
b=a;
}
Input is valid!!
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ _
```



```
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ ./a.out
Enter expression:
while(a<10){
if(a==5){
b=a*a;
}
}
Input is valid!!
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals/Practical7$ _
```

Conclusion

Written grammar to control statements and Loop control in YACC.