Practical - 12

Aim : Write a program for below grammar using yacc tool

(a) {a^n b | n>0}

(b) {a^n b^n | n>0}

(c)Validate variable name expression

(d) Validate “if” statement and nested “if” statement.

Program:

(A) {a^n b | n>0}

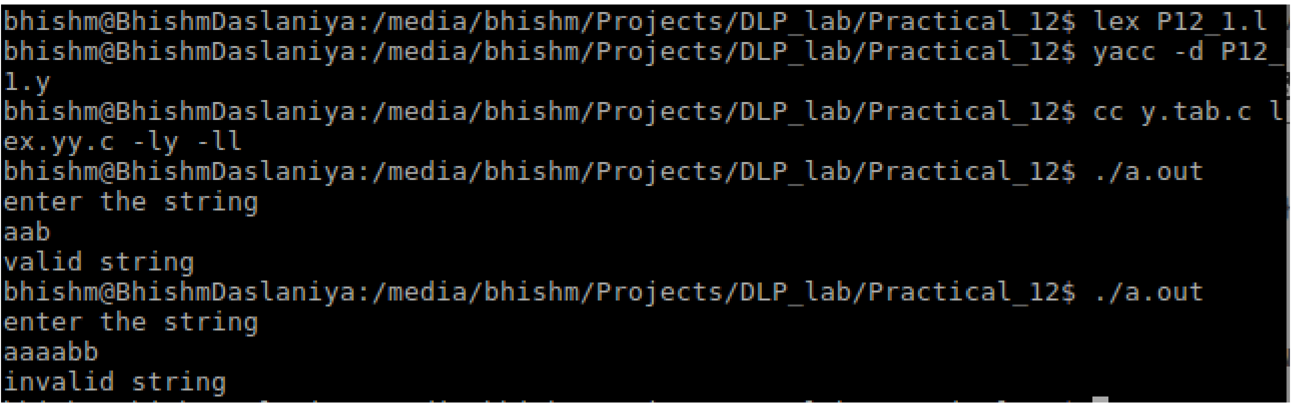
Lex Specification:

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

%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
```

Yacc Specification:

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

%token A B NL

%%
stmt: S B NL {printf("valid string\n");
            exit(0);}
;
S: A S | A
;
%%

int yyerror(char *msg)
{
printf("invalid string\n");
exit(0);
}

int main()
{
printf("enter the string\n");
yyparse();
}
```

Output:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ lex P12_1.l
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ yacc -d P12_
1.y
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ cc y.tab.c l
ex.yy.c -ly -ll
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the string
aab
valid string
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the string
aaaabb
invalid string
```

(B){a^n b^n | n>0}

Lex Specification:

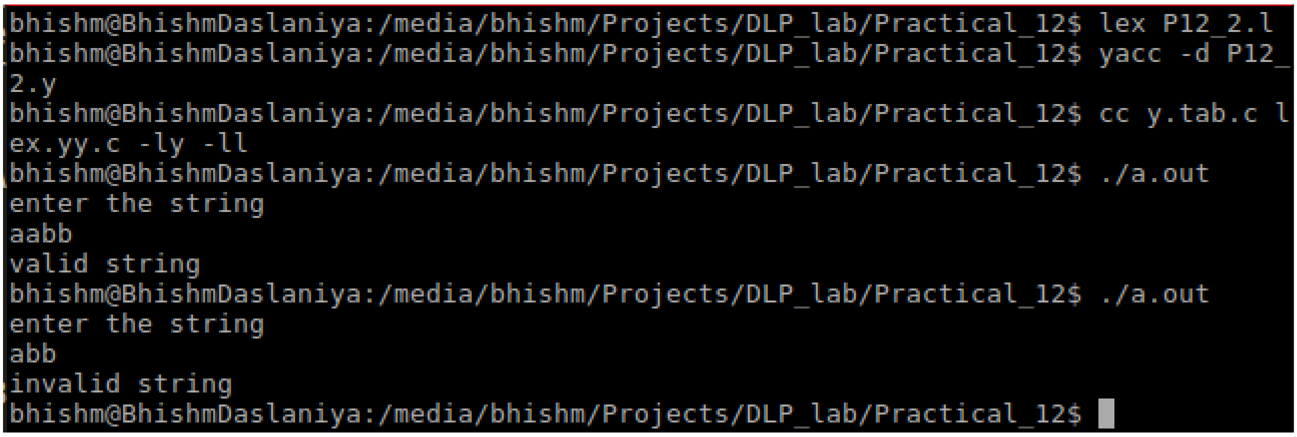
```
%{
#include "y.tab.h"
%}

%%
[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}
%%
```

Yacc Specification:

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

%token A B NL

%%
stmt: S NL {printf("valid string\n");
            exit(0);}
;
S: A S B | A B
```

```
;
%%

int yyerror(char *msg)
{
printf("invalid string\n");
exit(0);
}

int main()
{
printf("enter the string\n");
yyparse();
}
```

Output:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ lex P12_2.l
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ yacc -d P12_
2.y
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ cc y.tab.c l
ex.yy.c -ly -ll
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the string
aabb
valid string
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the string
abb
invalid string
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$
```

(C) Validate variable name expression

LEX Specification:

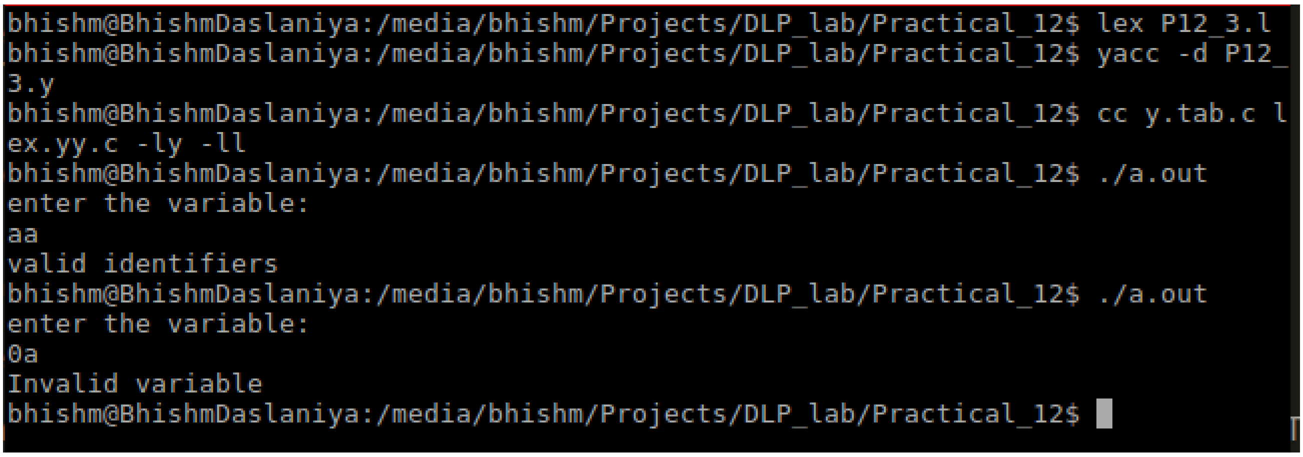
```
%{
#include"y.tab.h"
%}

%%
[a-zA-Z] {return LETTER;}
[0-9] {return DIGIT;}
[_] {return UND;}
[\n] {return NL;}
. {return yytext[0];}
%%
```

Yacc Specification:

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

%token DIGIT LETTER UND NL
```

```
%%
stmt: variable NL {printf("valid identifiers\n"); exit(0);}
;
variable: LETTER alphanumeric
;
alphanumeric: LETTER alphanumeric | DIGIT alphanumeric | UND alphanumeric | LETTER | DIGIT | UND
;

%%

int yyerror(char *msg)
{
 printf("Invalid variable\n");
 exit(0);
}
int main()
{
 printf("enter the variable: \n");
 yyparse();
}
```

Output:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ lex P12_3.l
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ yacc -d P12_
3.y
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ cc y.tab.c l
ex.yy.c -ly -ll
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the variable:
aa
valid identifiers
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the variable:
0a
Invalid variable
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$
```

(D) Validate “if” statement and nested “if” statement.

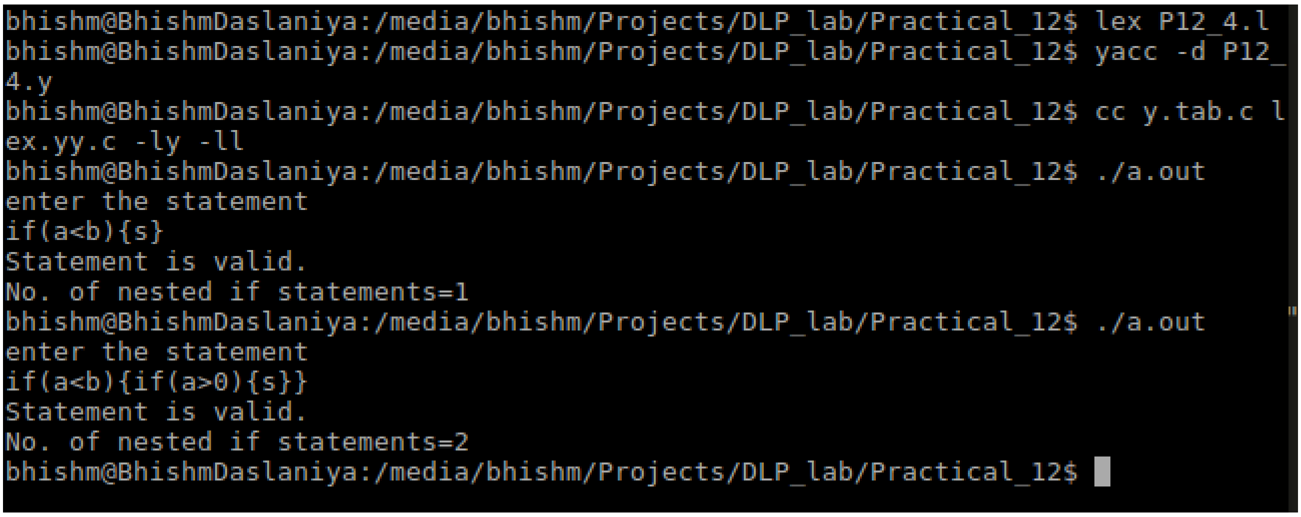
LEX Specification:

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

%%
"if" {return IF;}
[sS][0-9]* {return S;}
"<"|">"|"=="|"<="|">="|"!=" {return RELOP;}
[0-9]+ {return NUMBER;}
[a-z][a-zA-Z0-9_]* {return ID;}
\n {return NL;}
. {return yytext[0];}
%%
```

Yacc Specification:

```
%{
#include<stdio.h>
#include<stdlib.h>
int yylex();
int yyerror(char *s);
int count=0;
%}

%token IF RELOP S NUMBER ID NL

%%
stmt: if_stmt NL {printf("Statement is valid.\nNo. of nested if statements=%d\n",count);exit(0);}
;
if_stmt : IF'('cond')''{'if_stmt'}' {count++;} | S
;
cond: x RELOP x
;
x:ID | NUMBER
;
%%

int yyerror(char *msg)
{
printf("the statement is invalid\n");
exit(0);
}

int main()
{
printf("enter the statement\n");
yyparse();
}
```

Output:



```
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ lex P12_4.l
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ yacc -d P12_
4.y
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ cc y.tab.c l
ex.yy.c -ly -ll
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the statement
if(a<b){s}
Statement is valid.
No. of nested if statements=1
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$ ./a.out
enter the statement
if(a<b){if(a>0){s}}
Statement is valid.
No. of nested if statements=2
bhishm@BhishmDaslaniya:/media/bhishm/Projects/DLP_lab/Practical_12$
```

Conclusion: From this practical I have learnt about how to implement basic compiler program with yacc tool.