Practical : 8

Aim: **Write a program to convert infix to postfix using lex and YACC.**

Code: **Lex file: p8.l**

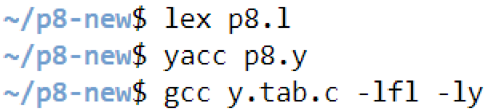
```
%{
#include<stdio.h>
#include "y.tab.h"
extern int yylval;
%}

%%

[0-9]+ {
 yylval=atoi(yytext);
  return NUMBER;
  }

[\t] ;
[\n] return 0;
. return yytext[0];

%%

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

YACC file: p8.y

```
%{
  /* Definition section */
  #include <stdio.h>
  #include <stdlib.h>
  #include "y.tab.h"
%}
%token    ID
%left    '+' '-'
%left    '*' '/'
```

```
%left    UMINUS
/* Rule Section */
%%
S  :  E
E  :  E'+'{A1();}T{A2();}
   |  E'-'{A1();}T{A2();}
   |  T
   ;
T  :  T'*'{A1();}F{A2();}
   |  T'/'{A1();}F{A2();}
   |  F
   ;
F  :  '('E{A2();}')'
   |  '-'{A1();}F{A2();}
   |  ID{A3();}
   ;
%%
#include"lex.yy.c"
char st[100];
int top=0;
//driver code
int main()
{
    printf("Enter infix expression:  ");
    yyparse();
    printf("\n");
    return 0;
}
A1()
{
    st[top++]=yytext[0];
}
A2()
{
    printf("%c", st[--top]);
}
A3()
{
    printf("%c", yytext[0]);
}}
```

Output:



```
~/p8-new$ lex p8.l
~/p8-new$ yacc p8.y
~/p8-new$ gcc y.tab.c -lfl -ly
```