

LAB 8

Alok

220101048

Sec- 'A'

1. Write a YACC program for converting infix to postfix expression.

Parser Source Code :

```
% {  
    /* Definition section */  
    #include <stdio.h>  
    #include <stdlib.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()
```

```
{
```

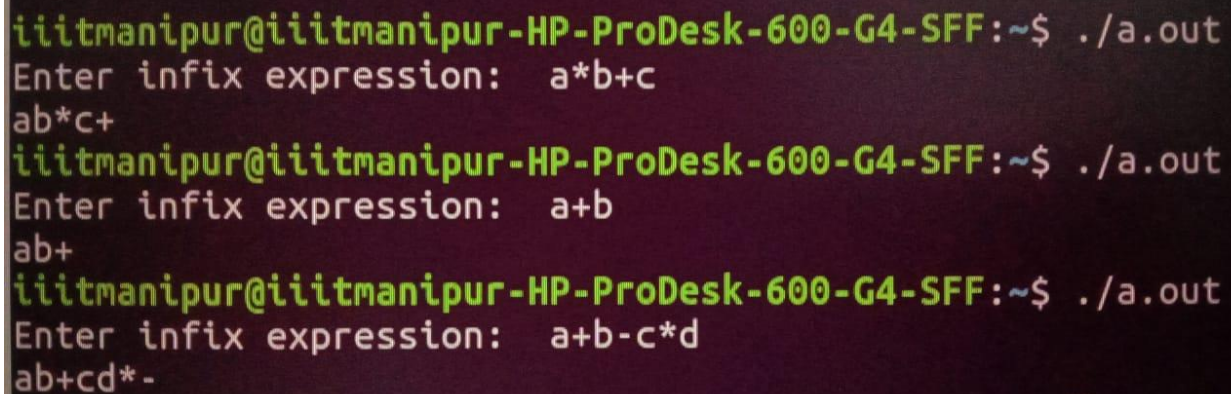
```
    printf("%c", yytext[0]);
```

```
}
```

Lexical Analyzer Source Code :

```
% {  
    /* Definition section */  
%}  
ALPHA [A-Z a-z]  
DIGIT [0-9]  
  
/* Rule Section */  
%%  
{ALPHA}({ALPHA}|{DIGIT})* return ID;  
{DIGIT}+          {yylval=atoi(yytext); return ID;}  
[\n \t]            yyterminate();  
.  
return yytext[0];  
%%
```

OUTPUT:



```
iitmanipur@iitmanipur-HP-ProDesk-600-G4-SFF:~$ ./a.out  
Enter infix expression:  a*b+c  
ab*c+  
iitmanipur@iitmanipur-HP-ProDesk-600-G4-SFF:~$ ./a.out  
Enter infix expression:  a+b  
ab+  
iitmanipur@iitmanipur-HP-ProDesk-600-G4-SFF:~$ ./a.out  
Enter infix expression:  a+b-c*d  
ab+cd*-
```