LPCC Assignment 5

Name: Digvijay Pawar
Class: TY.Btech Comp B2
Gr.No: 21810344
Roll No: 322043

Aim: Write a program for converting a simple expression into 3 address code.

Objective:

1. To understand LEX and YACC Concepts.
2. To implement LEX Program and Corresponding YACC program.
3. To study about Lex and yacc specification.
4. To study working of 3 address code generation.

Theory:

Three address code is a type of intermediate code which is easy to generate and can be easily converted to machine code.It makes use of at most three addresses and one operator to represent an expression and the value computed at each instruction is stored in temporary variable generated by compiler. The compiler decides the order of operation given by three address code.

General representation :–   a = b op c

Where a, b or c represents operands like names, constants or compiler generated temporaries and op represents the operator

There are 3 representations of three address code namely

1.Quadruple

2.Triples

3.Indirect Triples

Code:

1. File - yacc file(ass5.y)

```
%{
#include <stdio.h>
void yyerror(char*);
int yylex(void);
char n ='A';
int i=0;
%}

%token NUM
%token ID
%left '+''-''*''/'
%nonassoc UMINUS
%%

S : E {printf("x=%c\n",n-1);}

;

E:       NUM{}
```

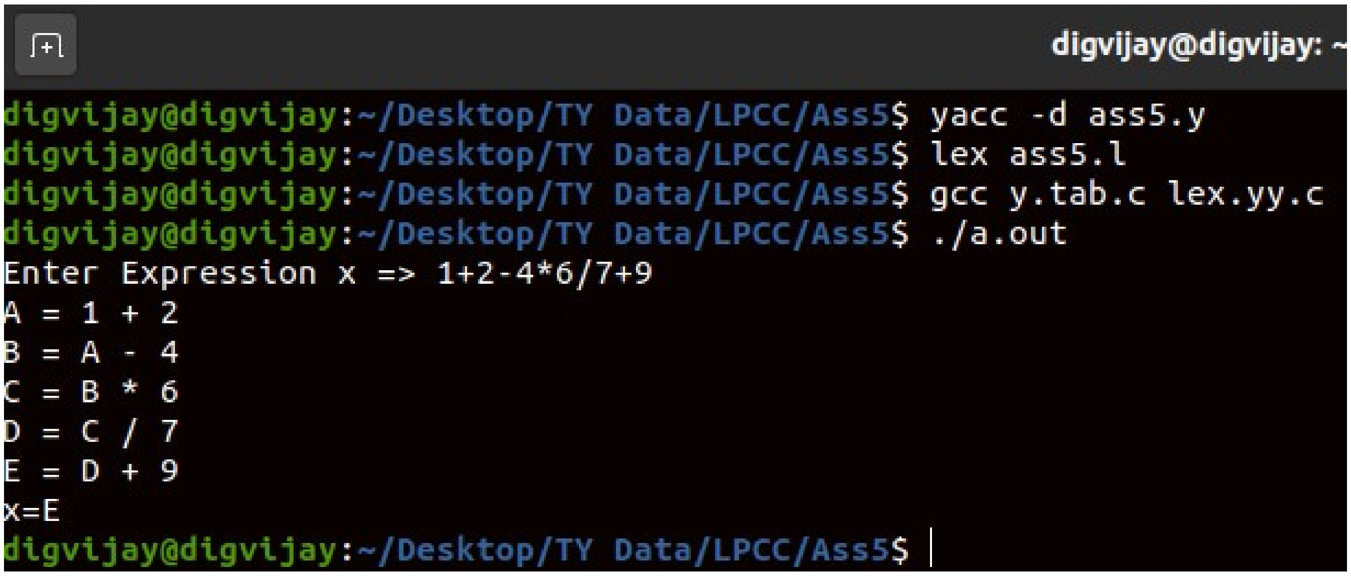
```
        |E'+'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'+',$3); n++;i++;}
        else{ printf("%c = %c %c %d\n",n,n-1,'+',$3);n++;}}
        |E'-'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'-',$3); n++;i++;}
        else{ printf("%c = %c %c %d\n",n,n-1,'-',$3);n++;}}
        |E'*'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'*',$3); n++;i++;}
        else{ printf("%c = %c %c %d\n",n,n-1,'*',$3);n++;}}
        |E'/'E{if(i==0){ printf("%c = %d %c %d\n",n,$1,'/',$3); n++;i++;}
        else{ printf("%c = %c %c %d\n",n,n-1,'/',$3);n++;}}
;

%%
void yyerror(char* s)
{
        fprintf(stderr,"%s\n",s);
}

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

        printf("Enter Expression x => ");
        yyparse();
        yywrap();
        return 0;
}
```

2. File - lex file(ass5.l)

```
 %{

 #include "y.tab.h"

 extern int yylval;

 %}
```

```
%%
[A-Za-z]([A-Za-z][0-9])* return ID;
[0-9]+ {yylval=atoi(yytext);return NUM;}
. {return yytext[0];}
\n {return 0;}
%%
```

Output :



```
digvijay@digvijay: ~
digvijay@digvijay:~/Desktop/TY Data/LPCC/Ass5$ yacc -d ass5.y
digvijay@digvijay:~/Desktop/TY Data/LPCC/Ass5$ lex ass5.l
digvijay@digvijay:~/Desktop/TY Data/LPCC/Ass5$ gcc y.tab.c lex.yy.c
digvijay@digvijay:~/Desktop/TY Data/LPCC/Ass5$ ./a.out
Enter Expression x => 1+2-4*6/7+9
A = 1 + 2
B = A - 4
C = B * 6
D = C / 7
E = D + 9
x=E
digvijay@digvijay:~/Desktop/TY Data/LPCC/Ass5$
```