Practical 5

Name: Rajatkumar Patel
Roll No.: 18BCE191

Aim

To implement a calculator in YACC.

Code

Input

Practical 5.l file

```
%{
#include <string.h>
#include <stdio.h>
#include "y.tab.h"
int lookup(char str[]);
int count=0;
char name[20][8];
float value[20]={0};


%}
%%
[0-9]+  {yylval.ival=atoi(yytext); return NUMBER;}
[0-9]*"."[0-9]+ {yylval.fval=atof(yytext); return FNUMBER;}
[-+%(){};=/*]   {return yytext[0];}
([_a-zA-Z][0-9]*)+ {yylval.ival=lookup(yytext);return ID;}
%%

int lookup(char str[]){
    int i;
    for( i=0;i<count;i++){
        if(strcmp(name[i],str)==0)
            return i;
    }
    count++;
    strcpy(name[i],str);
    return i;
}
int yywrap() {
 return 1;
}
```

Practical 5.y

```
%{

#include <stdio.h>
#include <stdlib.h>
void yyerror();
void yywrap();
extern float value[20];
int flag=0;

%}

%union
{
    int ival;
    float fval;
}

%token <ival> NUMBER
%token <ival> ID
%token <fval> FNUMBER
%type <fval> E

%right '='
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'

%%
       ArithmeticExpression:E';'{
              printf("result = %f",$1);
              return 0;
       };
       E:E'+'E {$$=$1+$3;}
       |E'-'E {$$=$1-$3;}
       |E'*'E {$$=$1*$3;}
       |E'/'E {$$=$1/$3;}
       |E'%'E {$$=(int)$1 % (int)$3;}
       |'('E')' {$$=$2;}
       |NUMBER      {$$=$1;}
       |FNUMBER {$$=$1;}
       ;

%%
int main(){
```

```
    printf("\nEnter any Arithmetic Expression: ");
    yyparse();
    if(flag==0)
    {
        printf("\nEntered Arithmetic Expression is Valid");
    }
}
void yyerror()
{
    printf("\nEntered Arithmetic Expression is Invalid");
    flag=1;
}
```

Output

```
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ flex Practical\ 5.l
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ yacc -d Practical\ 5.y
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ gcc lex.yy.c y.tab.c -w
(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ ./a.out

Enter any Arithmetic Expression: 1.3*1.3;
result = 1.690000
Entered Arithmetic Expression is Valid(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ ./a.out

Enter any Arithmetic Expression: (1+2.3434+(5.239+2.123)*(1.234))

;
result = 12.428107
Entered Arithmetic Expression is Valid(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$ ./a.out

Enter any Arithmetic Expression: 5.23*(1+2+3*5.34/21.12);
result = 19.657074
Entered Arithmetic Expression is Valid(base) rajat@rajat-VivoBook-S14-X430UA:/Rajat1/Books/Compiler Construction/Practicals$
```

Conclusion

Implemented a calculator using YACC and understood how parsing works.