```
%{
/* Definition section */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>

int flag = 0;
int yylex(void);
void yyerror(const char *s);

%}

%union {
        double dval;
}


%token <dval> NUMBER
%type <dval> Input ArithmeticExpression E


%left '+' '-'
%left '*' '/' '%'
%left UMINUS
%left '(' ')'

%start Input

%%

Input:
    ArithmeticExpression '\n' {
        printf("\nResult = %g\n", $1);
        return 0;
    }
;

ArithmeticExpression:
    E
;

E:
    E '+' E { $$ = $1 + $3; }
    | E '-' E { $$ = $1 - $3; }
    | E '*' E { $$ = $1 * $3; }
    | E '/' E {
        if ($3 == 0) {
            yyerror("Division by zero");
            $$ = 0;
```

```
        } else {
            $$ = $1 / $3;
        }
    }
    | E '%' E {
        if ($3 == 0) {
            yyerror("Division by zero");
            $$ = 0;
        } else {
            $$ = fmod($1, $3);
        }
    }
    | '(' E ')' { $$ = $2; }
    | NUMBER { $$ = $1; }
;

%%

// Driver code
int main()
{
    printf("Enter any arithmetic expression which can have operations Addition,
Subtraction, Multiplication, Division, Modulus, and Round brackets:\n");
    yyparse();
    if(flag == 0)
        printf("\nEntered arithmetic expression is valid\n\n");
    return 0;
}

void yyerror(const char *s)
{
    printf("\nEntered arithmetic expression is invalid: %s\n", s);
    flag = 1;
}
```