

desk_calculator.l

```
%{
#include <stdio.h>
#include <stdlib.h>

int paren_count = 0;
int error_flag = 0;
%}

DIGIT      [0-9]
NUMBER     (DIGIT)+(\.{DIGIT})?

%%

"+"        { printf("PLUS "); }
"-"        { printf("MINUS "); }
"*"        { printf("MULTIPLY "); }
"/"        { printf("DIVIDE "); }

"("        { printf("LPAREN "); paren_count++; }
")"        { printf("RPAREN "); paren_count--; if (paren_count < 0) error_flag = 1; }

{NUMBER}   { printf("NUMBER "); }

[ \t\n]+   { /* ignore spaces */ }

.          { printf("\nINVALID TOKEN: %s\n", yytext); error_flag = 1; }

%%

int yywrap() {
    return 1;
}

void check_expression_validity() {
    if (error_flag || paren_count != 0)
        printf("\nInvalid arithmetic expression.\n");
    else
        printf("\nValid arithmetic expression.\n");
}
}
```

sample.c

```
#include <stdio.h>
#include <string.h>

int yylex();
void check_expression_validity();
extern void yy_scan_string(const char*);

int main() {
    char input[1024];

    printf("Enter an arithmetic expression:\n");
    fgets(input, sizeof(input), stdin);
    input[strcspn(input, "\n")] = '\0';

    yy_scan_string(input);
    yylex();
    check_expression_validity();

    return 0;
}
```

OUTPUT

```
[cdlab88@localhost desk_calculator]$ vi desk_calculator.1
[cdlab88@localhost desk_calculator]$ vi sample.c
[cdlab88@localhost desk_calculator]$ lex desk_calculator.1
[cdlab88@localhost desk_calculator]$ gcc lex.yy.c sample.c -o desk_calculator
[cdlab88@localhost desk_calculator]$ ./desk_calculator
Enter an arithmetic expression:
(5+3)*2.3
LPAREN NUMBER PLUS NUMBER RPAREN MULTIPLY NUMBER
Valid arithmetic expression.
[cdlab88@localhost desk_calculator]$ ./desk_calculator
Enter an arithmetic expression:
(5+3))*2.3
LPAREN NUMBER PLUS NUMBER RPAREN RPAREN MULTIPLY NUMBER
Invalid arithmetic expression.
[cdlab88@localhost desk_calculator]$ ./desk_calculator
Enter an arithmetic expression:
(3+a)*2
LPAREN NUMBER PLUS
INVALID TOKEN: a
RPAREN MULTIPLY NUMBER
Invalid arithmetic expression.
```