

```

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

int num1, num2, num3;
char operator1, operator2;
}%

%%
[0-9]+      {
    if (num1 == 0) {
        num1 = atoi(yytext);
    } else if (num2 == 0) {
        num2 = atoi(yytext);
    } else {
        num3 = atoi(yytext);
    }
}

"+"         { if (!operator1) operator1 = '+'; else operator2 = '+'; }
"_"         { if (!operator1) operator1 = '-'; else operator2 = '-'; }
"*"         { if (!operator1) operator1 = '*'; else operator2 = '*'; }
"/"         { if (!operator1) operator1 = '/'; else operator2 = '/'; }
"%"         { if (!operator1) operator1 = '%'; else operator2 = '%'; }

\n          { return 0; } // End of expression

[ \t]       ; // Ignore spaces and tabs

.           { printf("Invalid character: %s\n", yytext); }

%%

int main(void) {
    printf("Enter an arithmetic expression with two operators only: ");
    yylex(); // Start the lexical analysis

    int result1 = 0, result2 = 0, final_result = 0;

    // Perform the first operation
    switch(operator1) {
        case '+': result1 = num1 + num2; break;
        case '-': result1 = num1 - num2; break;
        case '*': result1 = num1 * num2; break;
        case '/':
            if (num2 != 0)
                result1 = num1 / num2;
            else {
                printf("Error: Division by zero\n");
                return 1;
            }
    }
}

```

```

    }
    break;
case '%':
    if (num2 != 0)
        result1 = num1 % num2;
    else {
        printf("Error: Division by zero\n");
        return 1;
    }
    break;
default: printf("Invalid operator\n"); return 1;
}

```

// Perform the second operation

```

switch(operator2) {
    case '+': final_result = result1 + num3; break;
    case '-': final_result = result1 - num3; break;
    case '*': final_result = result1 * num3; break;
    case '/':
        if (num3 != 0)
            final_result = result1 / num3;
        else {
            printf("Error: Division by zero\n");
            return 1;
        }
        break;
    case '%':
        if (num3 != 0)
            final_result = result1 % num3;
        else {
            printf("Error: Division by zero\n");
            return 1;
        }
        break;
    default: final_result = result1; // In case there's no second operator
}

```

```

printf("Result: %d\n", final_result);
return 0;

```

```

}

```

```

int yywrap() {
    return 1;
}

```