# Practical-7

Lex code :

%{

#include <stdio.h>

%}

%%

.\*com$ { printf("Matched: %s", yytext); }

.\* { /\* ignore other lines \*/ }

%%

int main() {

yylex();

return 0;

}

int yywrap() {

return 1;

}

YACC code :

%{

#include <stdio.h>

#include <stdlib.h>

#define MAXSTACK 100

int stack[MAXSTACK];

int top = -1;

void push(int val) {

if (top >= MAXSTACK - 1) {

printf("Stack overflow\n");

exit(1);

}

stack[++top] = val;

}

int pop() {

if (top < 0) {

printf("Stack underflow\n");

exit(1);

}

return stack[top--];

}

int yylex(void);

int yyerror(char \*s);

%}

%token NUMBER

%%

input:

input line

|

;

line:

expr '\n' { printf("Result: %d\n", pop()); }

;

expr:

expr NUMBER { push($2); }

|

expr '+' { int b = pop(); int a = pop(); push(a + b); }

|

expr '-' { int b = pop(); int a = pop(); push(a - b); }

|

expr '\*' { int b = pop(); int a = pop(); push(a \* b); }

|

expr '/' { int b = pop(); int a = pop(); push(a / b); }

|

NUMBER { push($1); }

;

%%

int yyerror(char \*s) {

fprintf(stderr, "Error: %s\n", s);

return 0;

}

int main() {

printf("Enter postfix expression:\n");

yyparse();

return 0;

}





