```
#include <stdio.h>
#include <ctype.h>
char expr[100];
int i = 0;
// Forward declarations
void E();
void T();
void F();
// Grammar:
// E → T {('+' | '-') T}*
// T → F {('*' | '/') F}*
// F → '(' E ')' | id | number
void E() {
   T();
   while (expr[i] == '+' || expr[i] == '-') {
      char op = expr[i++];
      T();
      printf("%c", op);   // semantic action for + or -
   }
}
void T() {
   F();
   while (expr[i] == '*' || expr[i] == '/') {
      char op = expr[i++];
      F();
      printf("%c", op);   // semantic action for * or /
```

```
    }
}
void F() {
    // If it's a number (multi-digit)
    if (isdigit(expr[i])) {
        while (isdigit(expr[i])) {
            printf("%c", expr[i]);
            i++;
        }
        printf(" "); // space for multi-digit separation
    }
    // If it's a variable name
    else if (isalpha(expr[i])) {
        printf("%c", expr[i]);
        i++;
    }
    // If it's a parenthesized expression
    else if (expr[i] == '(') {
        i++;      // skip '('
        E();      // evaluate inside
        i++;      // skip ')'
    }
}
int main() {
    printf("Enter expression: ");
    scanf("%s", expr);
```

```
    printf("Postfix: ");
    E();
    printf("\n");
    return 0;
}
```