#include <stdio.h>  
#include <ctype.h>  
  
#define MAXSTACK 100  
#define POSTFIXSIZE 100  
  
  
int stack[MAXSTACK];  
int top = -1;  
void push(int item)  
{  
  
    if (top >= MAXSTACK - 1) {  
        printf("stack over flow");  
        return;  
    }  
    else {  
        top = top + 1;  
        stack[top] = item;  
    }  
}  
int pop(){  
    int item;  
    if (top < 0) {  
        printf("stack under flow");  
    }  
    else {  
        item = stack[top];  
        top = top - 1;  
        return item;  
    }  
}  
  
  
void EvalPostfix(char postfix[])  
{  
  
    int i;  
    char ch;  
    int val;  
    int A, B;  
  
  
    for (i = 0; postfix[i] != ')'; i++) {  
        ch = postfix[i];  
        if (isdigit(ch)) {  
  
            push(ch - '0');  
        }  
        else if (ch == '+' || ch == '-' || ch == '\*' || ch == '/') {  
  
            A = pop();  
            B = pop();  
  
            switch (ch)  
            {  
            case '\*':  
                val = B \* A;  
                break;  
  
            case '/':  
                val = B / A;

break;  
  
            case '+':  
                val = B + A;  
                break;  
  
            case '-':  
                val = B - A;  
                break;  
            }  
 push(val);  
        }  
    }  
    printf(" \n Result of expression evaluation : %d \n", pop());  
}  
  
int main()  
{  
  
    int i;  
  
  
    char postfix[POSTFIXSIZE];  
    printf("ASSUMPTION: There are only four operators(\*, /, +, -) in an expression and operand is single digit only.\n");  
    printf(" \nEnter postfix expression,\npress right parenthesis ')' for end expression : ");  
  
  
    for (i = 0; i <= POSTFIXSIZE - 1; i++) {  
        scanf("%c", &postfix[i]);  
  
        if (postfix[i] == ')')  
        {  
            break;  
        }  
    }  
  
  
  
    EvalPostfix(postfix);  
  
    return 0;  
}