

```

/*****
Athul Sabu
12
2025-07-23
*****/
/*****
ATHUL SABU
ROLL NO 20
14-07-2025
*****/
*****/
#include<stdio.h>
#include<ctype.h>
#define SIZE 100
char stack[SIZE];
int TOP = -1;
void push(char c){
    if (TOP ==SIZE-1)
        return;
    else {
        TOP++;
        stack[TOP] = c;
    }
}
char pop(){
    if (TOP ==-1)
        return '\0';
    else{
        char c = stack[TOP];
        TOP--;
        return c;
    }
}
int precedence(char c){
    if (c=='^') return 3;
    else if (c=='*' || c=='/') return 2;
    else if (c=='+' || c=='-') return 1;
    else return 0;
}
void infixToPostfix(char infix[], char postfix[]){
    char ch ;
    int i = 0, j=0;
    for(i=0; infix[i] !='\0'; i++){
        ch = infix[i];
        if(isdigit(ch))
            postfix[j++] = ch;
        else if (ch=='('){
            push(ch);
        }
        else if (ch ==')'){
            char poppedChar = pop();
            while(poppedChar !='('){
                postfix[j++] =poppedChar;
                poppedChar = pop();
            }
        }
        else {
            while( TOP!=-1 && precedence(stack[TOP])
                ==precedence(ch)){
                postfix[j++] = pop();
            }
            push(ch);
        }
    }
}

```

```

while(TOP !=-1){
    postfix[j++] =pop();
}
postfix[j] = '\0';
}
int evalPostfix(char postfix[]){
    int S2[32];
    int top =-1;
    int i =0;
    int a,b;
    char ch;
    while((ch=postfix[i++]) != '\0'){
        if (isdigit(ch)){
            S2[++top] = ch - '0';
        }
        else{
            a = S2[top--];
            b = S2[top--];
            switch(ch){
                case '+': S2[++top] = b+a; break;
                case '-': S2[++top] = b-a; break;
                case '*': S2[++top] = b*a; break;
                case '/': S2[++top] = b/a; break;
            }
        }
    }
    int result = S2[top];
    return result;
}
void main(){
    char postfix[SIZE] ,infix[SIZE];
    printf("Enter the infix value (don't enter white space) : ");
    scanf("%s", infix);
    infixToPostfix(infix, postfix);
    printf("\nPostfix exp : %s\n", postfix);
    int result = evalPostfix(postfix);
    printf("result = %d\n", result);
}
/*****OUTPUT*****/
*****
Enter the infix value (don't enter white space) : (3+2)*5
Postfix exp : 32+5*
result = 25
*****/
/*****OUTPUT*****/
*****
Compiled successfully. Press ■ Run.
Enter the infix value (don't enter white space) : (3+3)*3(3+3)*3
(3+3)*3

Postfix exp : 33+3*
result = 18
*/

```