```c
#include <stdio.h>
#include <ctype.h>

// Character stack for infix to postfix conversion
char charStack[100];
int charTop = -1;

void pushChar(char x) {
    charStack[++charTop] = x;
}

char popChar() {
    if (charTop == -1)
        return -1;
    else
        return charStack[charTop--];
}

int priority(char x) {
    if (x == '(')
        return 0;
    if (x == '+' || x == '-')
        return 1;
    if (x == '*' || x == '/')
        return 2;
    return -1;
}

// Integer stack for postfix evaluation
int intStack[100];
int intTop = -1;

void pushInt(int val) {
    intStack[++intTop] = val;
}

int popInt() {
    return intStack[intTop--];
}

int evaluatePostfix(char* expr) {
    for (int i = 0; expr[i] != '\0'; i++) {
        if (isdigit(expr[i])) {
            pushInt(expr[i] - '0');
        } else {
            int b = popInt();
            int a = popInt();
            switch (expr[i]) {
                case '+': pushInt(a + b); break;
                case '-': pushInt(a - b); break;
                case '*': pushInt(a * b); break;
                case '/': pushInt(a / b); break;
            }
        }
    }
    return popInt();
}

int main() {
    char exp[100];
    char postfix[100];
    char *e, x;
    int i = 0;

    printf("Enter the infix :: ");
    scanf("%s", exp);

    printf("\nPostfix expression: ");
    e = exp;
    while (*e != '\0') {
        if (isalnum(*e)) {
            printf("%c", *e);
            postfix[i++] = *e;
        } else if (*e == '(') {
            pushChar(*e);
        } else if (*e == ')') {
            while ((x = popChar()) != '(') {
                printf("%c", x);
                postfix[i++] = x;
            }
        } else {
            while (charTop != -1 && priority(charStack[charTop]) >= priority(*e)) {
                x = popChar();
                printf("%c", x);
                postfix[i++] = x;
            }
            pushChar(*e);
        }
        e++;
    }
    while (charTop != -1) {
        x = popChar();
        printf("%c", x);
        postfix[i++] = x;
    }
    postfix[i] = '\0';

    printf("\nEvaluated result: %d\n", evaluatePostfix(postfix));
    return 0;
}
```