```c
#include <stdio.h>
#define MAX 100

char stack[MAX];
int top = -1;

void push(char c) {
    if (top < MAX - 1) {
        stack[++top] = c;
    }
}

char pop() {
    if (top >= 0) {
        return stack[top--];
    } else {
        return '\0';
    }
}

int precedence(char c) {
    if (c == '+' || c == '-') {
        return 1;
    } else if (c == '*' || c == '/') {
        return 2;
    }
    return 0;
}
```

```c
void infixtopostfix(char infix[], char postfix[]) {
    int i = 0, j = 0;
    char token;

    while ((token = infix[i]) != '\0') {
        if (isOperand(token)) {
            postfix[j++] = token;
        } else if (token == '(') {
            push(token);
        } else if (token == ')') {
            while (top != -1 && stack[top] != '(') {
                postfix[j++] = pop();
            }
            pop();
        } else {
            while (top != -1 && precedence(stack[top]) >= precedence(token)) {
                postfix[j++] = pop();
            }
            push(token);
        }
        i++;
    }


    while (top != -1) {
        postfix[j++] = pop();
    }
    postfix[j] = '\0';
}

int main() {
    char infix[MAX], postfix[MAX];
    printf("Enter infix expression: ");
    scanf("%s", infix);
    infixtopostfix(infix, postfix);
    printf("Postfix expression: %s\n", postfix);
    return 0;
}
```

```
Enter infix expression: (a+(b*c)+(d/e))
Postfix expression: abc*+de/+

Process returned 0 (0x0)   execution time : 37.305 s
Press any key to continue.
```