Experiment 04\infix_to_postfix.c

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

#define MAX 100

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

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

char pop() {
    if (top == -1)
        return -1;
    return stack[top--];
}

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

void infixToPostfix(char* infix, char* postfix) {
    int i = 0, j = 0;
    char x;
    push('(');
    strcat(infix, ")");
    while (infix[i]) {
        if (isalnum(infix[i])) {
            postfix[j++] = infix[i];
        } else if (infix[i] == '(') {
            push(infix[i]);
        } else if (infix[i] == ')') {
            while ((x = pop()) != '(') {
                postfix[j++] = x;
            }
        } else {
            while (priority(stack[top]) >= priority(infix[i])) {
                postfix[j++] = pop();
            }
            push(infix[i]);
```

```c
        }
        i++;
    }
    postfix[j] = '\0';
}

int evalPostfix(char* postfix) {
    int stackVal[MAX];
    int topVal = -1;
    int i = 0;
    while (postfix[i]) {
        if (isdigit(postfix[i])) {
            stackVal[++topVal] = postfix[i] - '0';
        } else {
            int val2 = stackVal[topVal--];
            int val1 = stackVal[topVal--];
            switch(postfix[i]) {
                case '+': stackVal[++topVal] = val1 + val2; break;
                case '-': stackVal[++topVal] = val1 - val2; break;
                case '*': stackVal[++topVal] = val1 * val2; break;
                case '/': stackVal[++topVal] = val1 / val2; break;
            }
        }
        i++;
    }
    return stackVal[topVal];
}

int main() {
    char infix[MAX], postfix[MAX];
    printf("Enter infix expression (single digit operands): ");
    scanf("%s", infix);
    infixToPostfix(infix, postfix);
    printf("Postfix expression: %s\n", postfix);
    int result = evalPostfix(postfix);
    printf("Evaluation result: %d\n", result);
    return 0;
}

```