

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

#define MAX 100

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

void push(char c) {
    stack[++top] = c;
}

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

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[MAX];
    int i, j = 0;
    char x;
    char token;

    for (i = 0; infix[i] != '\0'; i++) {
        token = infix[i];

        if (isalnum(token)) {
            postfix[j++] = token;
        }
        else if (token == '(') {
            push(token);
        }
        else if (token == ')') {
            while ((x = pop()) != '(')
                postfix[j++] = x;
        }
    }
    postfix[j] = '\0';
}
```

```

        postfix[j++] = x;
    }
    else {
        while (precedence(stack[top]) >= precedence(token))
            postfix[j++] = pop();
        push(token);
    }
}

while (top != -1)
    postfix[j++] = pop();

postfix[j] = '\0';
printf("Postfix Expression: %s\n", postfix);
}

int main() {
    char infix[MAX];

    printf("Enter Infix Expression: ");
    scanf("%s", infix);

    infixToPostfix(infix);

    return 0;
}

```

```
PS C:\Users\Admin\Desktop\ROHANA\stackpgm1.c> cd ~\c:\Us
ixtopostfix } ; if ($?) { .\infixtopostfix }
Enter Infix Expression: (a+b)-c/d
Postfix Expression: ab+cd/-
PS C:\Users\Admin\Desktop\ROHANA\stackpgm1.c> 
```