

Q Program to convert infix to postfix

```
#include <stdio.h>      #include <string.h>
#define N 30             #include <ctype.h>
char stack[N];
int top = -1;
```

```
void push (char x) {
    if (top == N-1) {
        printf("Stack Overflow");
    }
    else {
        top++;
        stack[top] = x;
    }
}
```

```
char pop () {
    if (top == -1) {
        printf("Stack Underflow");
    }
    else {
        char item = stack[top];
        top--;
    }
    return item;
}
```

```
int precedence (char x) {
```

```
char peek () {
    if (top == -1) {
        printf("Stack Underflow");
    }
}
```

```
return stack[top];
```

```
return
```

```
}
```

```
switch (x) {
```

```
    case '+':
```

```
        return 1;
```

```
    case '-':
```

```
        return 1;
```

```
    case '*':
```

```
        return 2;
```

```
    case '/':
```

```
        return 2;
```

```
    case '^':
```

```
        return 3;
```

```
    default:
```

```
        break
```

```
}
```

```
int isright
```

```
int isOk(char x) {
```

```
    return ch == '+' || ch == '-' || ch == '*' || ch == '/' || ch == '^';
```

```
}
```

```
void infix
```

```
void itb
```

```
void itb(char infix[], char postfix[]) {
```

```
    int i = 0, int j = 0;
```

```
    char ch;
```

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

```
        ch = infix[i]
```

```
        if (ch == 'c') {
```

```
            push(ch);
```

```
        }
```

```
        else if (ch == '(') {
```

```
            while (top != -1 && peek() != 'c') {
```

```
                postfix[j++] = pop();
```

```
            }
```

```
            postfix[j] = pop();
```


else if (isalnum(ch)) {

postfix[j++] = ch;

}

else if (isop(ch)) {

while (top != -1 && peek != '(')

if (precedence(ch) < precedence(peek()))

precedence(ch) == precedence(peek()) &&

! isright(ch))

{

postfix[j++] = pop();

}

push(ch)

}

}

while (top != -1) {

postfix[j++] = pop();

}

postfix[j] = '\0';

}

int main() {

char infix[N], postfix[N];

printf("Enter expression:");

scanf("%s", infix);

itoa(infix, postfix);

printf(postfix);

}

M4
6/10/25

o/r

Enter expression: A+B*(C+D)

ABC++D+

Enter expression: (A+B)*C-(D/E)*F

ABC*DEF/-G+N*+

M4