

Q Program to convert infix to postfix

```
# include <stdio.h>      # include <string.h>
# define N 30              # include <ctype.h>
int 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];
}
```

```

switch (x) {
    case '+':
        return 1;
    case '-':
        return 1;
    case '*':
        return 2;
    case '/':
        return 2;
    case '^':
        return 3;
    default:
        break;
}

isright (char x) {
    return x == '^';
}

```

~~it is right~~

```
int isOkC(char x) {  
    return ch == '+'  
}
```

~~void infix()~~

~~if it's~~

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

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

Chau Ch

```
for (i = 0; infix[i] != '10'; i++) {
```

ch = infis[i]

if (ch == 'c') {

push (ch)

45

else if (ch == ')') {

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

postfix [i++] = bot();

```
    pop();  
};
```

else if (isalnum(ch)) {

postfix[j+1] = ch;

}

else if (isok(ch)) {

while (tok != -1 && peek != '(' &&

precedence(ch) < precedence(peek)) ||

(precedence(ch) == precedence(peek) &&
! isright(ch)))

}

postfix[j+1] = tok();

}

push(ch)

}

}

while (tok != -1) {

postfix[j+1] = tok();

}

postfix[i] = '0';

}

int main()

char infix[N], postfix[N];

printf("Enter expression: ");

scanf("%s", infix);

itoa(infix, postfix);

printf(postfix);

3/6/10/25

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

ABCDEF^G^H^I

MR