

LAB 3 [INFIX - POSTFIX]

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

```
int F (char symbol)
```

```
{
```

```
    switch (symbol)
```

```
    {
```

```
        case '+':
```

```
        case '-': return 2;
```

```
        case '*':
```

```
        case '/': return 4;
```

```
        case '^':
```

```
        case '$': return 5;
```

```
        case 'c': return 0;
```

```
        case '#': return -1;
```

```
        default: return 8;
```

```
    }
```

```
}
```

```
int G (char symbol)
```

```
{
```

```
    switch (symbol)
```

```
    {
```

```
        case '+':
```

```
        case '-': return 1;
```

```
        case '*':
```

```

    case '/': return 3;
    case '^':
    case '$': return 6;
    case '(': return 9;
    case ')': return 0;
    default: return 7;
}

```

}

```

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

```

```

    int top, i, j;
    char s[20], symbol;
    top = -1;
    s[++top] = '#';
    j = 0;
    for (i = 0; i < strlen(infix); i++)
    {

```

```

        symbol = infix[i];
        while (F(s[top]) > G(symbol))
        {

```

```

            postfix[j] = s[top--];

```

```

            while (F(s[top]) > G(symbol)) j++;

```

}

```

        if (F(s[top]) != G(symbol))

```

```

            s[++top] = symbol;

```

else

```

        j = Top++;

```



```
while (s[top] != '#')
{
    postfix[j++] = s[top--];
}
postfix[j] = '\0';
}

void main
{
    char infix[20];
    char str postfix[20];
    printf("Enter the valid infix expression:\n");
    scanf("%s", infix);
    infix -> postfix (infix, postfix);
    printf("The postfix expression is:\n");
    printf("%s\n", postfix);
    getch();
}
```