

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

int F (char symbol) {
    switch (symbol) {
        case '+': return 1;
        case '-': return 2;
        case '*': return 3;
        case '/': return 4;
        case '^': return 5;
        case ')': return 6;
        case '#': return -1;
        default: return 8;
    }
}
```

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

```
    switch (symbol) {
        case '+': return 1;
        case '-': return 2;
        case '*': return 3;
        case '/': return 4;
        case '^': return 5;
        case '$': return 6;
        case '(': return 7;
        case ')': return 0;
        default: return 8;
    }
}
```

```
void infix postfix (char infix[], char postfix[]).
{
    int i, j;
    char s[30], 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--];
            j++;
        }
        if (F(s[top]) != g(symbol))
            s[++top] = symbol;
        else
            top--;
    }
    while (s[top] != '#');
    {
        postfix[j+1] = s[top--];
    }
    postfix[j] = '\0';
}
void main().
{
    char infix[20];
    char postfix[20];
    clrscr();
    printf("enter the valid infix expression");
    scanf("%s", infix);
```

classmate
Date _____
Page _____

infix postfix (infix, postfix);
printf ("The postfix exp. is %h");
printf ("%s \n", postfix);
getch();

```
Last login: Wed Sep 30 12:08:17 on ttys000
Nishchals-MacBook-Pro:~ nishchal$ /var/folders/qg/drk3d
/T/geany_run_script_21T4R0.sh ; exit;
Enter infix expression:
((a+b)*c-(d-e))^(f+g)
The postfix expression is as follows
ab+c*de--fg+^
```

```
(program exited with code: 0)
Press return to continue
```

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

int F(char symbol) {
    switch(symbol) {
        case '+': return 1;
        case '-': return 2;
        case '*': return 3;
        case '/': return 4;
        case '^': return 5;
        case '$': return 6;
        case '(': return 0;
        case ')': return -1;
        default : return 8;
    }
}

int G(char symbol) {
    switch(symbol) {
        case '+': return 1;
        case '-': return 2;
        case '*': return 3;
        case '/': return 4;
        case '^': return 5;
        case '$': return 6;
        case '(': return 7;
        case ')': return 8;
        default : return 9;
    }
}

void infix_postfix(char infix[], char postfix[]) {
    int top,i,j;
    char s[30], 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--];
            j++;
        }
        if(F(s[top]) != G(symbol)) {
            s[++top] = symbol;
        }
    }
}
```

```

        case '*': return 3;
        case '/': return 3;
        case '^': return 4;
        case '$': return 6;
        case '(': return 9;
        case ')': return 0;
    default : return 7;
}
}

void infix_postfix(char infix[], char postfix[]) {
    int top,i,j;
    char s[30], 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--];
            j++;
        }
        if(F(s[top]) != G(symbol)) {
            s[++top] = symbol;
        }
        else {
            top--;
        }
    }
    while(s[top]!='#') {
        postfix[j++]=s[top--];
    }
    postfix[j] = '\0';
}

int main() {
    char infix[20];
    char postfix[20];
    printf("Enter infix expression:\n");
    scanf("%s",infix);
    infix_postfix(infix,postfix);
    printf("The postfix expression is as follows \n");
    printf("%s\n",postfix);
}

```