

# LAB - 3

WAP to convert valid parenthesized infix arithmetic expression to postfix expression.

```
#include <stdio.h>
#include <string.h>
#include <process.h>
int F (char symbol)
{
    switch (symbol)
    {
        case '+':
        case '-': return (2);
        case '*':
        case '/': return (4);
        case '^':
        case '$': return (5);
        case '(': 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 (1);
        case ')': return (0);
        default: return (-1);
    }
}

```

```

void infix-postfix (char infix[], char postfix[])
{
    int top = -1, i, j = 0;
    char s[30], symbol;
    s[++top] = '#';
    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';
}

```



```
void main()
```

```
{
```

```
    char infix[20];
```

```
    char postfix[20];
```

```
    printf("Enter infix expression \n");
```

```
    scanf("%s", &infix);
```

```
    infix_postfix(infix, postfix);
```

```
    printf("Postfix expression : %s", postfix);
```

```
    getch();
```

```
}
```