

```
#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;
```

Teacher's Signature : \_\_\_\_\_

```

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

```

```

}

```

```

}

```

```

void infixPostfix ( 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++;
        }

```

Teacher's Signature : \_\_\_\_\_



```
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 some infix expression : ");  
    scanf ("%s", infix);
```

```
    infixPostfix (infix, postfix);
```

```
    printf ("The postfix of above express  
            is : ");
```

```
    printf ("%s\n", postfix);  
    getch(); }
```

Teacher's Signature : \_\_\_\_\_

## OUTPUT

write some infix expression :  $(a + b - c) * d$

the postfix is :  $a + b c - ) d * )$  ,

---

enter some infix expression : (a+)b-c)\*d)  
the postfix of above is: a+bc-)\*d\*)