

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

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

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

```
#include <ctype.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 'C': return 9;
```

```
        case 'J': return 0;
```

```
        default: return 7;
```

```
    }
```

```
}
```



```
void infix_postfix (char infix [], char postfix []) {
```

```
    int top, j, i;
```

```
    char s[30];
```

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

```
    int k, a = 0, b = 0, flag = 0;
```

```
    printf ("Enter valid infix exp.\n");
```

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

```
    for (k = 0; k < strlen(infix); k++)
```

```
    }
```



```
if (isalnum(infix[k]) && isalnum(infix[k+1]))
```

```
flag++
```

```
if (infix[k] == '(')
```

```
a++;
```

```
if (infix[k] == ')')
```

```
b++;
```

```
}
```

```
if (a != b || flag != 0)
```

```
{
```

```
printf("Invalid infix exp!");
```

```
exit(0);
```

```
}
```

```
infix-postfix(infix, postfix);
```

```
printf("The postfix exp is: \n");
```

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

```
return 0;
```

```
}
```