

C-Program to convert Infix expression to Postfix expression:-

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

```
#include <string.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 2;
```

```

case 'n':
case 'x': return 6;
case '(': return 9;
case ')': return 0;
default : return 7;
}

```

```

}
void infix_Postfix (char infix[], char Postfix[])
{
    int top, i, j;
    char stack[30], symbol;
    top = -1;
    stack[++top] = '#';
    j = 0;
    for (i = 0; i < strlen(infix); i++)
    {
        symbol = infix[i];
        while (F(stack[top]) > G(symbol))
        {
            Postfix[j] = stack[top--];
            j++;
        }
        if (F(stack[top]) != G(symbol))
            stack[++top] = symbol;
        else
            top--;
    }
}

```

```

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

int main()
{
    char infix[20];
    char postfix[20];
    printf("Enter the infix expression from user:\n");
    scanf("%s", infix);
    infix -> postfix(infix, postfix);
    printf("The postfix expression is : \n");
    printf("%s\n", postfix);
    return 0;
}

```