

# 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 3;
```



```
case 'n':
```

```
case '&': 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;
}
```



- one.c
- prime.c
- q8.c
- q10.c
- smallest.c
- ~\De...JAVA)
- array.java
- Roots.java
- ~\Des...folder
- a.c
- array.c
- b.c
- bin.c
- cal.c
- calculator.c
- cie.c
- count.c
- dd.c
- ex.java
- exa.java
- hash.c
- index.html
- matpro.c
- new.c
- oo.c
- pass by.c
- prec.c
- pri.c
- prime.c
- rand func.c

```

1  #include<stdio.h>
2  #include<string.h>
3  int F(char symbol)
4  {
5      switch(symbol)
6      {
7          case '+':
8          case '-':return 2;
9          case '*':
10         case '/':return 4;
11         case '^':
12         case '$':return 5;
13         case '(':return 0;
14         case '#': return -1;
15         default : return 8;
16     }
17 }
18 int G(char symbol)
19 {
20     switch(symbol)
21     {
22         case '+':
23         case '-':return 1;
24         case '*':
25         case '/':return 3;
26         case '^':
27         case '&':return 6;
28         case '(':return 9;
29         case ')':return 0;
30         default:return 7;
31     }
32 }
33 void infix_postfix(char infix[],char postfix[])
34 {
35     int top,j,i;
36     char Stack[30],symbol;
37     top=-1;
38     Stack[++top]='#';

```

Status gcc -Wall -c "prec.c" (in directory: C:\Users\DELL\Desktop\New folder)

Compiler Compilation finished successfully.

Messages

Scribble

line: 22 / 70 col: 0 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: G



one.c

prime.c

q8.c

q10.c

smallest.c

~\De...JAVA)

array.java

Roots.java

~\Des...folder

a.c

array.c

b.c

bin.c

cal.c

calculator.c

cie.c

count.c

dd.c

ex.java

exa.java

hash.c

• index.html

matpro.c

new.c

oo.c

pass by.c

prec.c

pri.c

prime.c

rand func.c

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

```
{
    int top,j,i;
    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;
}
```

Status gcc -Wall -c "prec.c" (in directory: C:\Users\DELL\Desktop\New folder)

Compiler Compilation finished successfully.

Messages

Scribble



C:\WINDOWS\SYSTEM32\cmd.exe

Enter the infix expression from user:

$a^b * c - d + e / f / (g + h)$

The postfix expression is :

$ab^c * d - ef / gh + / +$

-----  
(program exited with code: 0)

Press any key to continue . . .