infixtopostfix\_07.cpp

**Compile:** g++ infixtopostfix\_07.cpp -o infixtopostfix\_07

**Run:** ./infixtopostfix\_07

**Program:**

#include<iostream>

#include<cstring>

#include<stack>

using namespace std;

intgetWeight(char ch) {

switch (ch) {

case '/':

case '\*': return 2;

case '+':

case '-': return 1;

default : return 0;

}

}

void infix2postfix(char infix[], char postfix[], int size) {

stack<char> s;

int weight;

inti = 0;

int k = 0;

charch;

while (i< size) {

ch = infix[i];

if (ch == '(') {

s.push(ch);

i++;

continue;

}

if (ch == ')') {

while (!s.empty() &&s.top() != '(') {

postfix[k++] = s.top();

s.pop();

}

if (!s.empty()) {

s.pop();

}

i++;

continue;

}

weight = getWeight(ch);

if (weight == 0) {

postfix[k++] = ch;

}

else {

if (s.empty()) {

s.push(ch);

}

else {

while (!s.empty() &&s.top() != '(' &&

weight<= getWeight(s.top())) {

postfix[k++] = s.top();

s.pop();

}

s.push(ch);

}

}

i++;

}

while (!s.empty()) {

postfix[k++] = s.top();

s.pop();

}

postfix[k] = 0;

}

int main() {

char infix[] = "A+(B\*C)/D";

int size = strlen(infix);

char postfix[size];

infix2postfix(infix,postfix,size);

cout<<"\nInfix Expression :: "<<infix;

cout<<"\nPostfix Expression :: "<<postfix;

cout<<endl;

return 0;

}

**Output:**

