**Experiment No.3c (OpenEnded Experiment)**

**Title:** Translation from Prefix to Postfix using Recursion

**Problem Statement:** Write a C++ program to convert prefix to postfix expression using the recursion.

**Algorithm:**

**Step 1** : Read the Prefix expression in reverse order (from right to left)

**Step 2:** If the symbol is an operand, then push it onto the Stack

**Step 3**: If the symbol is an operator, then pop two operands from the Stack  
**Step 4**: Create a string by concatenating the two operands and the operator after them.  
**string = operand1 + operand2 + operator**  
**Step 5:** And push the resultant string back to Stack

**Step 6:** Repeat the above steps until end of Prefix expression.

**Code:**

#include<iostream>

#include<stack>

using namespace std;

string prefix;

string postfix;

int i,l;

stack<string> s;

bool isoperator(char ch)

{

switch(ch)

{

case'+':

case'-':

case'\*':

case'/':

return true;

}

return false;

}

void pretopost(int i)

{

if(i<0)

{

postfix=s.top();s.pop();

cout<<"Postfix expression:"<<postfix<<endl;

return;

}

if(isoperator(prefix[i]))

{

string p1=s.top();s.pop();

string p2=s.top();s.pop();

string temp=p1+p2+prefix[i];

s.push(temp);

}

else{

s.push(string(1,prefix[i]));

}

i=i-1;

pretopost(i);

}

int main()

{

cout<<"Enter the prefix expression:";

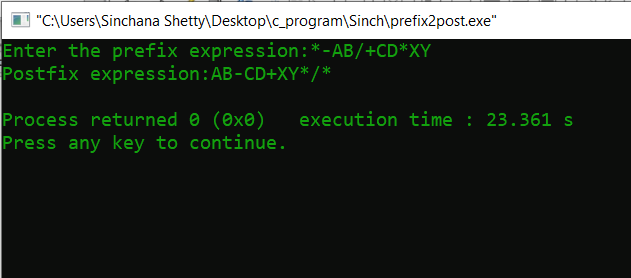
cin>>prefix;

l=prefix.size();

pretopost(l-1);

}

**Result:**



**Analysis(Limitations):**

In this code there is Conversion of Prefix expression directly to Postfix without going through the process of converting them first to Infix and then to Postfix is much better in terms of computation and better understanding the expression.