**Experiment No. 3C**

**Title :** Implementation of prefix to postfix expression using recursion.

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

**Algorithm:**

**Step 1:** Start

**Step 2:** Declare strings named prefix and postfix and a stack of strings

**Step 3:** Input the prefix expression

**Step 4:** Read the Prefix expression in reverse order (from right to left)

**Step 5**: If the symbol is an operand, then push it onto the Stack

**Step 6:** If the symbol is an operator, then pop two operands from the Stack

**Step 7:** Create a string by concatenating the two operands and the operator after them.

**Step 8:**  string = operand1 + operand2 + operator

And push the resultant string back to Stack

**Step 9:** Repeat the above steps until end of Prefix expression. And then display

**Step 10:** Stop

**Program :**

#include<iostream>

#include<stack>

using namespace std;

//Declare strings for prefix and postfix expressions

string prefix;

string postfix;

//Declare a stack of type string using stack library function

stack<string> s;

bool isoperator(char ch)//checks if character is a operator

{

switch(ch)

{

case '+' :

case '-' :

case '\*' :

case '/' :

return true;

}

return false;

}

void convert(int i)

{

if(i<0)//exit condition after completion of conversion

{

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

cout<<"\nPostfix expression :"<<postfix<<endl; //print postfix expression

return;

}

if(isoperator(prefix[i]))

{

//if it is an operator pop the strings and concatinate

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])); //push in stack if not an operator

}

i = i - 1;//controls the exit condition

convert(i);//recursive call

}

int main()

{

cout<<"\*\*\*\* P R E F I X T O P O S T F I X\*\*\*\*\n ";

cout<<"enter the prefix expression : ";

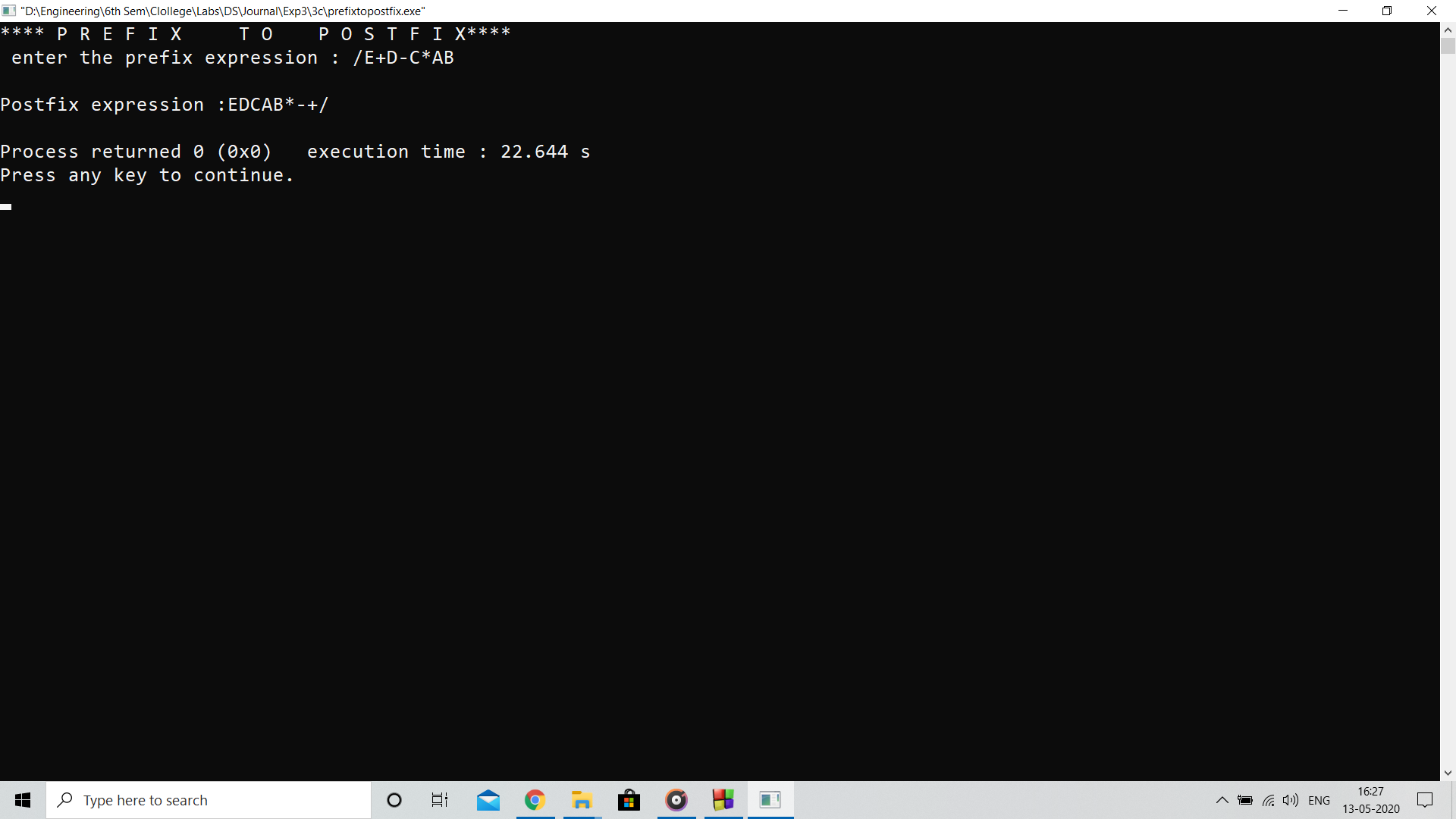
cin>>prefix;//input prefix expression

int l = prefix.size();//get length of prefix expression and store in L

convert(l-1);//convert the expression to postfix

}

**Output:**

****

**Analysis:**

Program uses stack data structure of strings to convert the prefix expression to postfix expression using recursive function call

**Limitation:**

Program fails to work with parenthesis and power operator if present in prefix expression