Ex. No. 3A	CONVERSION OF INFIX TO POSTFIX EXPRESSION
26-07-2017	

Question:

Develop a C++ program to convert an infix expression to postfix expression using stack.

Algorithm:

1. Start.
2. Scan the Infix expression from left to right for tokens (Operators, Operands & Parentheses) and perform the steps 2 to 5 for each token in the Expression.
3. If token is operand, Append it in postfix expression.
4. If token is left parentheses "(", push it in stack.
5. If token is an operator,
   - Pop all the operators which are of higher or equal precedence then the incoming token and append them (in the same order) to the output Expression.
   - After popping out all such operators, push the new token on stack.
6. If ")" right parentheses are found,
   - Pop all the operators from the Stack and append them to Output String, till you encounter the Opening Parenthesis "(".
   - Pop the left parenthesis but don't append it to the output string (Postfix notation does not have brackets).
7. When all tokens of Infix expression have been scanned. Pop all the elements from the stack and append them to the Output String.
8. The Output string is the Corresponding Postfix Notation and display the postfix expression.
9. End.

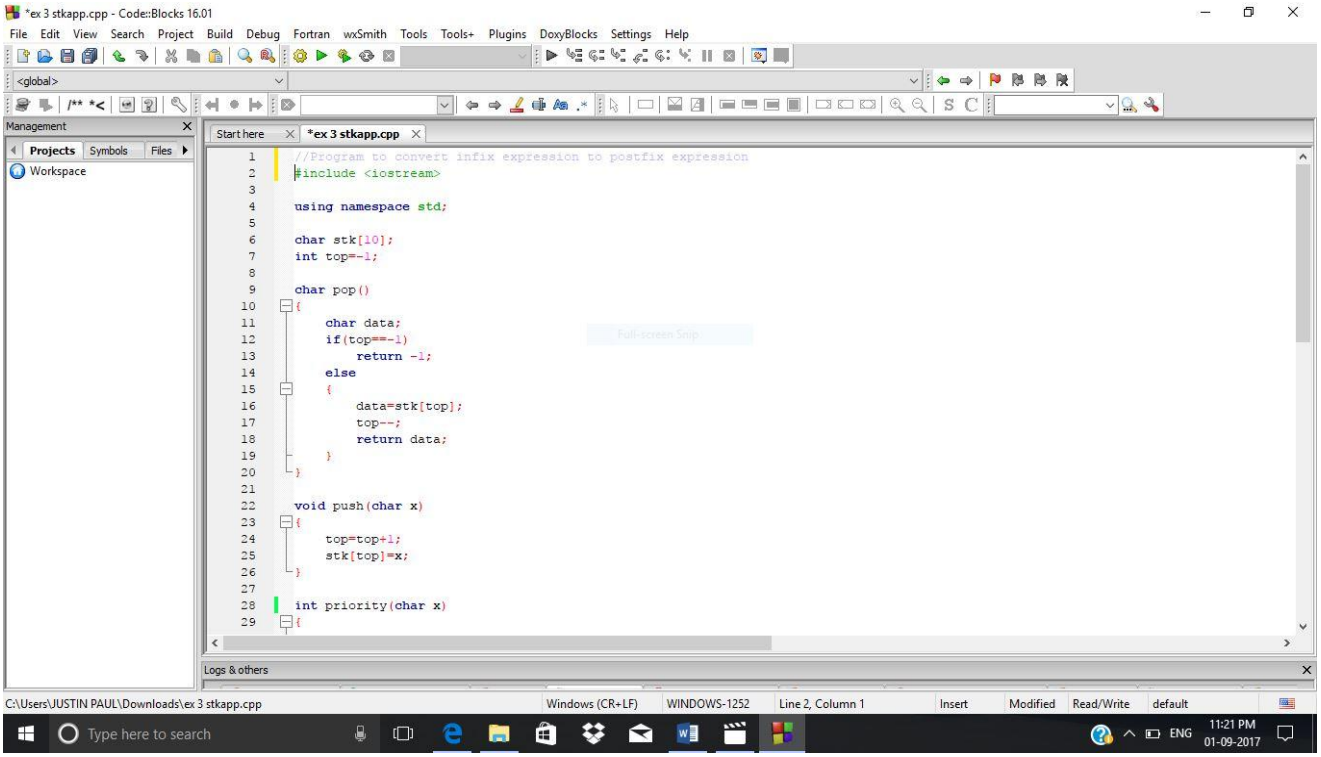
Program:

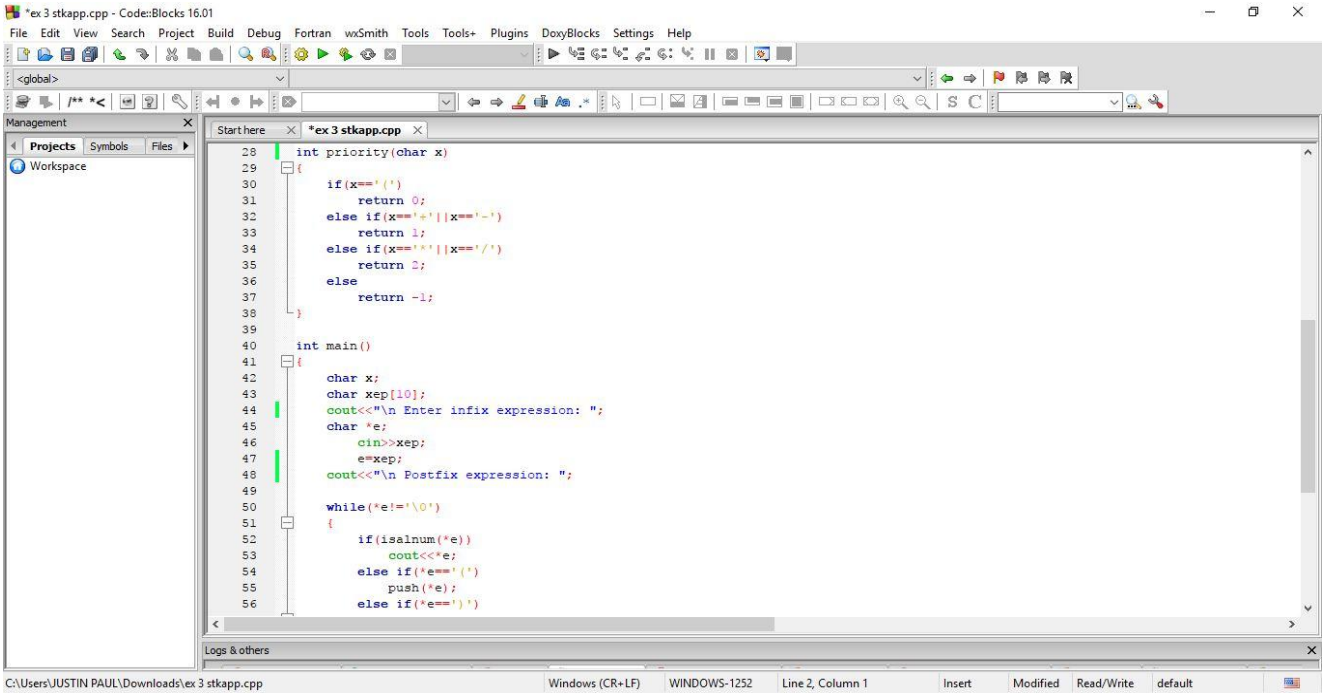
```
#include <iostream>
using namespace std;
char stk[10];
int top=-1;

char pop()
{
    char data;
    if(top==-1)
        return -1;
    else
    {
        data=stk[top];
        top--;
        return data;
    }
}
void push(char x)
{
    top=top+1;
    stk[top]=x;}
```

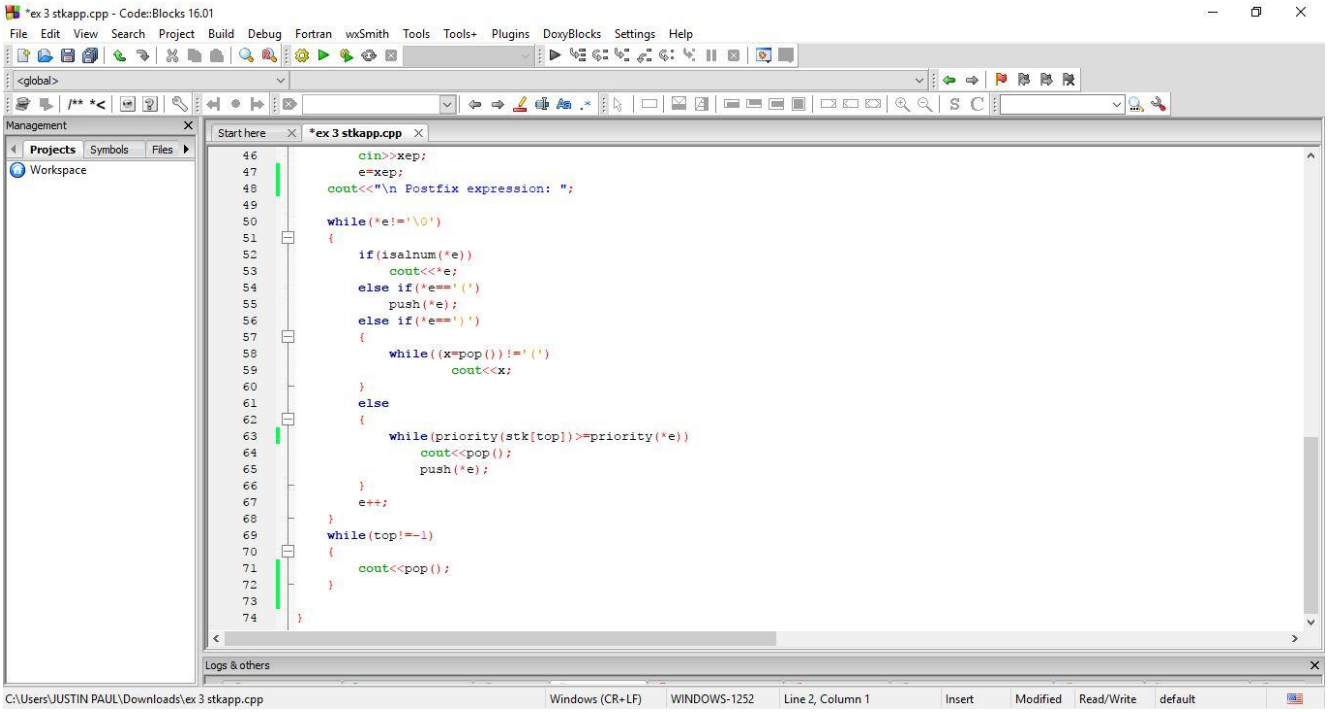
```
int priority(char x)
{
    if(x=='(')
        return 0;
    else if(x=='+'||x=='-')
        return 1;
    else if(x=='*'||x=='/')
        return 2;
    else
        return -1;
}

int main()
{
    char x;
    char xep[10];
    cout<<"\n Enter infix expression: ";
    char *e;
        cin>>xep;
        e=xep;
    cout<<"\n Postfix expression: ";
```

```
while(*e!='\0')
{
   if(isalnum(*e))
      cout<<*e;
   else if(*e=='(')
      push(*e);
   else if(*e==')')
   {
      while((x=pop())!='(')
            cout<<x;
   }
   else
   {
      while(priority(stk[top])>=priority(*e))
         cout<<pop();
         push(*e);
   }
   e++;
}
while(top!=-1)
{
   cout<<pop(); }}
```

Output:


*ex 3 stkapp.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
*ex 3 stkapp.cpp
//Program to convert infix expression to postfix expression
#include <iostream>
using namespace std;
char stk[10];
int top=-1;
char pop()
{
char data;
if(top==-1)
return -1;
else
{
data=stk[top];
top--;
return data;
}
}
void push(char x)
{
top=top+1;
stk[top]=x;
}
int priority(char x)
{
Logs & others
C:\Users\JUSTIN PAUL\Downloads\ex 3 stkapp.cpp
Windows (CR+LF)
WINDOWS-1252
Line 2, Column 1
Insert
Modified
Read/Write
default


*ex 3 stkapp.cpp - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
*ex 3 stkapp.cpp
int priority(char x)
{
if(x=='(')
return 0;
else if(x=='+'||x=='-')
return 1;
else if(x=='*'||x=='/')
return 2;
else
return -1;
}
int main()
{
char x;
char xep[10];
cout<<"\n Enter infix expression: ";
char *e;
cin>>xep;
e=xep;
cout<<"\n Postfix expression: ";
while(*e!='\0')
{
if(isalnum(*e))
cout<<*e;
else if(*e=='(')
push(*e);
else if(*e==')')
Logs & others
C:\Users\JUSTIN PAUL\Downloads\ex 3 stkapp.cpp
Windows (CR+LF)
WINDOWS-1252
Line 2, Column 1
Insert
Modified
Read/Write
default



VIDEO URL:

https://youtu.be/CdtwCYs8ID0

RESULT:

The program to convert infix expression to postfix expression using stack is implemented successfully and the output is verified.