```
#include<iostream>
#include<stdlib.h>
using namespace std;
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -



- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> class Stack
{                                       int max,top;
                                        T stack[100];
                        public:
                                        Stack();
                                        int isFull();
                                        int isEmpty();
                                        void push(T data);
                                        T pop();
};
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> Stack <T> :: Stack()
{
           max=99;
           top=0;
}
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> int Stack <T> :: isFull()
{
        if (top==max)           return 1;
        else                    return 0;
}
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> int Stack <T> :: isEmpty()
{
        if (top==0)     return 1;
        else            return 0;
}
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> void Stack <T> :: push(T data)
{
                top=top+1;
                stack[top]=data;
}
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
template <class T> T Stack <T> :: pop()
{
                T pdata;
                pdata=stack[top];
                top=top-1;
                return(pdata);
}
- ---------- ---------- ---------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- -
```

```
#include<ctype.h>
#include<string.h>

#include "Stack.h"

- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
                    Function to Compute "In-Stack Priority" of Operators
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
int isp(char c)
{
        int r;
        switch(c)
        {
                case '^':       r=4;
                                break;
                case '*':
                case '/':
                case '%':       r=3;
                                break;
                case '+':
                case '-':       r=2;
                                break;
                case '(':       r=1;
                                break;
                case '#':       r=0;
                                break;
        }
        return(r);
}
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
                    Function to Compute "In-Coming Priority" of Operators
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
int icp(char c)
{
        int r;
        switch(c)
        {
                case '^':       r=4;
                                break;
                case '*':
                case '/':
                case '%':       r=3;
                                break;
                case '+':
                case '-':       r=2;
                                break;
        }
        return(r);
}
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -


- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
main()
{               int ch;
                char infix[100];

                cout << ".....Enter the INFIX Expression ?..... ";
                cin >> infix;

                Stack <char> st;

                char token, x, y;

                st.push('#');
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -


- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
                cout << ".....Resulting Postfix Expression..... ";

                for (int i=0; infix[i]!='\0' ; ++i)
                {
                        token=infix[i];

                        if ( isalpha(token) )
                                                cout << token << " ";
                        else if (token=='(')
                                                st.push('(');
                        else if (token==')')
                        {
                                while ( (x=st.pop())!='(' )
                                {
                                        cout << x << " ";
                                }
                        }
                        else
                        {               x = st.pop();

                                        while ( isp(x) >= icp(token) )
                                        {
                                                cout << x << " ";
                                                x = st.pop();
                                        }
                                        st.push(x);
                                        st.push(token);
                        }
                }
                while ( !st.isEmpty() )
                {
                        cout << st.pop() << " ";
                }
                cout << endl;
}
- ---------- ---------- ---------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- -
```