Program Prefix

```
#include<iostream>
#include<stack>
#include<locale> //for function isalnum()
#include<algorithm>
using namespace std;

int preced(char ch)
{
      if(ch == '+' || ch == '-')
      {
            return 1;
      }
      else if(ch == '*' || ch == '/')
      {
                  return 2;
      }
      else if(ch == '^')
      {
                  return 3;
      }
      else
      {
                  return 0;
      }
}

string inToPost(string infix)
{
      stack<char> stk;
      stk.push('#');
      string postfix = "";
      string::iterator it;
      for(it = infix.begin(); it!=infix.end(); it++)
      {
                  if(isalnum(char(*it)))
                  postfix += *it;
                  else if(*it == '(')
                  stk.push('(');
                  else if(*it == '^')
                  stk.push('^');
                  else if(*it == ')')
                  {
                  while(stk.top() != '#' && stk.top() != '(')
                  {
                              postfix += stk.top();
                              stk.pop();
                  }
                     stk.pop();
            }
            else
            {
                  if(preced(*it) > preced(stk.top()))
                              stk.push(*it);
                  else
                  {
                              while(stk.top() != '#' && preced(*it) <=
preced(stk.top()))
                              {
                              postfix += stk.top();
                              stk.pop();
```

```
                                }
                                stk.push(*it);
                        }
                        }
        }
        while(stk.top() != '#')
        {
                        postfix += stk.top();
                        stk.pop();
        }
        return postfix;
}

string inToPre(string infix)
{
        string prefix;
        reverse(infix.begin(), infix.end());
        string::iterator it;
        for(it = infix.begin(); it != infix.end(); it++)
        {
                        if(*it == '(')
                        *it = ')';
                        else if(*it == ')')
                        *it = '(';
        }
        prefix = inToPost(infix);
        reverse(prefix.begin(), prefix.end());
        return prefix;
}

int main()
{
        char infix[10];
        cout<<"Enter the infix expression";
        cin>>infix;

        cout << "Prefix Form Is: " << inToPre(infix) << endl;
}
```

Output:

```
Enter the infix expression
(a+b)*c
Prefix Form Is: *+abc
```