```cpp
//Aditya Agre
//SYCOA06
//Infix to postfix

#include<iostream>
#include<string.h>
#include<stack>
using namespace std;
// A Function with return type int to return precedence of operators
int
precedence (char ch)
{
  if (ch == '^')
    return 3;
  else if (ch == '/' || ch == '*')
    return 2;
  else if (ch == '+' || ch == '-')
    return 1;
  else
    return -1;
}

// A Function to convert infix expression to postfix expression
string
infixToPostfix (string s)
{
  stack < char >st;            //For stack operations, we are using C++ built in stack
  string ans = "";
  for (int i = 0; i < s.length (); i++)
    {
      char ch = s[i];
// If the current character is an operand, add it to our answer string.
      if ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')
          || (ch >= '0' && ch <= '9'))
          ans += ch;              // Append the current character of string in our answer
// If the current character of string is an '(', push it to the stack.
      else if (ch == '(')
          st.push ('(');
// If the current character of string is an ')', append the top character of stack in our answer string
// and pop that top character from the stack until an '(' is encountered.
      else if (ch == ')')
          {
            while (st.top () != '(')
              {
```

```cpp
                ans += st.top ();    // Append the top character of stack in our answer
                st.pop ();
            }
            st.pop ();
        }
    //If an operator is scanned
        else
        {
            while (!st.empty () && precedence (s[i]) <= precedence (st.top ()))
            {
                ans += st.top ();
                st.pop ();
            }
            st.push (ch);            // Push the current character of string in stack
        }
    }
    // Pop all the remaining elements from the stack
    while (!st.empty ())
    {
        ans += st.top ();
        st.pop ();
    }
    return ans;
}

int
main ()
{
    string s;
    cin >> s;
    cout << infixToPostfix (s);
    return 0;
}
```

```
a+b+(c/d-k)*b
ab+cd/k-b*+


...Program finished with exit code 0
Press ENTER to exit console.
```