//Infix to Postfix

```
#include<iostream>
#include<stack>
#include<string>

using namespace std;
// Function to return precedence of operators
int prec(char c)
{
   if (c == '^')
   return 3;
   else if (c == '/' || c == '*')
   return 2;
   else if (c == '+' || c == '-')
   return 1;
   else
   return -1;
}

void infixToPostfix(string s)
{
stack<char> st;
string result;
for (int i = 0; i < s.length(); i++) {
   char c = s[i];

if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')
|| (c >= '0' && c <= '9'))
result += c;
```

```
// If the scanned character is an // ‘(‘, push it to the stack.
else if (c == '(')
st.push('(');
// If the scanned character is an ‘)’,
// pop and to output string from the stack
// until an ‘(‘ is encountered.
else if (c == ')')
{
   while (st.top() != '(')
   {
      result += st.top();
      st.pop();
}
st.pop();
}
// If an operator is scanned
else {
while (!st.empty() && prec(s[i]) <= prec(st.top())) {
   result += st.top();
   st.pop();

}
st.push(c);
}
}
// Pop all the remaining elements from the stack while (!st.empty()) { result += st.top(); st.pop(); }
cout << result << endl;
}
// Driver's code
int main() {
string exp = "a+b*(c^d-e)^(f+g*h)-i";
```

```
// Function call
infixToPostfix(exp);
return 0;

}
```