```
#include <bits/stdc++.h>

using namespace std;

bool is_operator()
{
}

string postfixToInfix(string str)
{
    stack<string> st;

    for (int i = 0; i < str.size(); i++)
    {
        if (str[i] >= 'A' && str[i] <= 'Z')
        {
            st.push(string(1, str[i]));
        }
        else
        {
            string x = st.top();
            st.pop();
            string y = st.top();
            st.pop();
            string m = '(' + y + str[i] + x + ')';
            st.push(m);
        }
    }
    return st.top();
}

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

void infixToPostfix(string s)
{
    stack<char> st;
    for (int i = 0; i < s.size(); i++)
    {
        if (s[i] >= 'A' && s[i] <= 'Z')
        {
            cout << s[i];
        }
        else if (s[i] == '(')
        {
            st.push(s[i]);
        }
        else if (s[i] == ')')
        {
            while (st.top() != '(')
            {
                cout << st.top();
                st.pop();
            }
            st.pop();
        }
        else
        {
            while (!st.empty() && st.top() != '(' && precedence(s[i]) >= precedence(st.top()))
            {
                cout << st.top();
                st.pop();
            }
            st.push(s[i]);
        }
    }
    while (!st.empty())
    {
        cout << st.top();
        st.pop();
    }
}

int main()
{
    string s;
    cin >> s;
    string x = postfixToInfix(s);
    infixToPostfix(x);
}

=========================
#include <bits/stdc++.h>
using namespace std;
vector<stack<char>> ans;
void rec(int ind, int size, string str1, stack<char> prev)
{
    if (ind == size)
    {
        ans.push_back(prev);
        return;
    }
    for (int i = 0; i < size; i++)
    {
        stack<char> new_stack = prev;
        new_stack.push(str1[i]);
        rec(ind + 1, size, str1, new_stack);
    }
}
int main()
{
    int n;
    cin >> n;
    string str1;
    cin >> str1;
    stack<char> new_stack;
    rec(0, n, str1, new_stack);
    for (int i = 0; i < ans.size(); i++)
    {
        while (ans[i].size() > 0)
        {
            cout << ans[i].top();
            ans[i].pop();
        }
        cout << endl;
    }
    return 0;
}
#include <bits/stdc++.h>
using namespace std;

int f(int a[], int low, int hi){
    if( low == hi ){
        return a[low];
    }
    int temp = f(a,low+1,hi);
    return min(temp, a[low]);
}
#include <bits/stdc++.h>
using namespace std;

int maxx(int a[], int n)
{
    int maxi = INT_MIN;
    for (int i = 0; i < n; i++)
    {
        maxi = max(maxi, a[i]);
    }
    return maxi;
}
int minii(int a[], int n)
{
    int mini = INT_MAX;
    for (int i = 0; i < n; i++)
    {
        mini = min(mini, a[i]);
    }
    return mini;
}

void countsort(int a[], int n)
{
    int min = abs(minii(a, n));
    int max = maxx(a, n);
    int tm[max + min + 1];
    for (int i = 0; i < max + min + 1; i++)
    {
        tm[i] = 0;
    }
    for (int i = 0; i < n; i++)
    {
        tm[a[i] + abs(min)]++;
    }
    for (int i = 0; i < max + min + 1; i++)
    {
        cout << tm[i] << " ";
    }
    cout << "FAAAA" << endl;
    for (int i = 0, j = 0; i < n;)
    {
        while (tm[j] > 0)
        {
            a[i] = j - abs(min);
            i++;
            tm[j]--;
        }
        j++;
    }
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
}

int main()
{
    int n = 11;
    int a[] = {5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5};

    countsort(a, n);
}
```