Problem(L2Q2): Input an expression and make a truth table for some set of values with postfix.

Program:

```
#include <iostream>

#include <stack>
#include <cmath>
#include <queue>
using namespace std;
int var = 0;
stack<char> st;
stack<int> st2;
deque<pair<char, int>> postfix;
deque<pair<char, int>> postfix2;
void printexp(deque<pair<char, int>> post)
{
    cout << "Postfix expression" << endl;
    for (auto i = post.cbegin(); i != post.end(); i++)
    {
        pair<char, int> it = *i;
        if (isalpha(it.first))
            var++;
        cout << it.first;
    }
}
int precedence(char c)
{
    if (c == '^')
        return 3;
    else if (c == '/' || c == '*')
        return 2;
    else if (c == '+' || c == '-')
        return 1;
    else
        return -1;
}
bool isOperand(char ch)
{
    return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') || (ch >= '0' && ch <= '9');
}
void infixToPostfix(string infix)
{
    int n = infix.size();
    for (int i = 0; i < n; i++)
    {
        if (isOperand(infix[i]))
        {
            postfix.push_back(make_pair(infix[i], 0));
        }
```

```
        else if (infix[i] == '(')
            st.push('(');
        else if (infix[i] == ')')
        {
            while (st.top() != '(')
            {
                postfix.push_back(make_pair(st.top(), 0));
                st.pop();
            }
            st.pop();
        }
        else
        {
            while (!st.empty() && st.top() != '(' && precedence(st.top()) >=
precedence(infix[i]))
            {
                postfix.push_back(make_pair(st.top(), 0));
                st.pop();
            }
            st.push(infix[i]);
        }
    }
    while (!st.empty())
    {
        postfix.push_back(make_pair(st.top(), 0));
        st.pop();
    }
}
void replacevar()
{
    for (auto i = postfix.cbegin(); i != postfix.end(); i++)
    {
        pair<char, int> it = *i;
        if (isalpha(it.first))
        {
            cin >> it.second;
            cout << it.second << " ";
        }
        postfix2.push_back(it);
    }
}

bool postcalc(deque<pair<char, int>> post)
{
    int num, a, b, c;
    bool h;
    while (!post.empty())
    {
        pair<char, int> it = post.front();
        if (isdigit(it.first))
```

```
        {
            num = it.first - 48;
            st2.push(num);
        }
        else if (isalpha(it.first))
        {
            st2.push(it.second);
        }
        else
        {
            a = st2.top();
            st2.pop();
            b = st2.top();
            st2.pop();
            switch (it.first)
            {
            case '+':
                c = a + b;
                break;
            case '-':
                c = b - a;
                break;
            case '*':
                c = a * b;
                break;
            case '/':
                c = b / a;
                break;
            case '^':
                c = pow(a, b);
                break;
            case '>':
                h = b > a;
                break;
            case '<':
                h = b < a;
                break;
            }
            st2.push(c);
        }
        post.pop_front();
    }
    return h;
}

int main()
{
    int many;
    string main;
    cout << "Enter your Condition" << endl;
```

```
    cin >> main;
    infixToPostfix(main);
    printexp(postfix);
    cout << "\nYou have " << var << " variables . "
         << "How many sets of solution" << endl;
    cin >> many;
    for (int i = 0; i < many; i++)
    {
        replacevar();
        cout << "     " << postcalc(postfix2) << endl;
    }
}
```

Input & Output

```
Enter your Condition
2*X+3*Y>0
Postfix expression
2X*3Y*+0>
You have 2 variables . How many sets of solution
2
1 3 5 -7
1 3      1
5 -7      0
PS C:\Users\Tonmo\Documents\New folder>
```