

```

#include <iostream>
#include <fstream>
#include <string>
#include <list>

using namespace std;

template <class T >
T teclado(list<char>& lista, T& it, char c, bool &ant) {

    if (c == '-') {

        ant = true;
        return lista.begin();
    }
    else if (c == '+') {

        return lista.end();
    }
    else if (c == '*') {

        if (it != lista.end()) ++it;
        return it;
    }
    else if (c == '3') {

        if (it != lista.end()) it = lista.erase(it);
        return it;
    }
    else {

        it = lista.insert(it, c);

        ++it;

        return it;
    }
}

bool resuelveCaso() {

    list<char> lista;

    auto it = lista.begin();

    string entrada;

    bool ant = false;

    getline(cin, entrada);

    if (!cin) return false;

```

```

    for (char c : entrada) {

        it = teclado(lista, it, c, ant);
    }

    for (auto it = lista.begin(); it != lista.end(); ++it) {
        cout << *it;
    }

    cout << endl;

    return true;
}

int main() {
# ifndef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); // save old buf and redirect std :: cin to
casos .txt
# endif

    while (resuelveCaso()) {}

# ifndef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system(" PAUSE ");
# endif

    return 0;
}

```