**Обмеження по пам'яті:** 64Мб

**Обмеження по часу:** 2с

Майнкрафтія відома не тільки своєю кубічністю. В цій країні діє неймовірна редстоун механіка, на основі якої можна будувати різні механізми. Зокрема, дана механіка дозволяє створювати логічні оператори, які представляють собою побітові операції. Логічні операції бувають чотирьох видів: **AND**(і, 2 вхідних сигнали), **OR**(або, 2 вхідних сигнали), **XOR**(додавання за модулем два, 2 вхідних сигнали), **NOT**(не, 1 вхідний сигнал).

Автор задачі побудував механізм на основі побітових операції, який представляє собою неорієнтоване дерево, в коренем в вершині 1. Всі листки дерева, крім кореня, є початковими сигналами і можуть повертати 0 або 1. Всі інші елементи є одними з побітових операцій. Такі елементи приймають сигнали від дітей і повертають значення результату функції, яку вони виконують. Результатом такого механізму буде значення, яке поверне корінь.

Вам стало цікаво, який буде результат механізму, якщо змінити сигнал, який повертає саме даний вхід, на протилежний.

**Вхідні дані:**

Перший рядок містить одне ціле число **n** – кількість вершин в дереві (входів+бітових операторів).

Наступні **n** рядків містять опис кожної з вершин, починаючи нумерацію з 1. Спочатку іде тип вершини: "AND", "OR", "XOR", "NOT" або "IN". Якщо ця вершина типу "IN" – це вхідний сигнал, тобто далі іде одне число, яке позначає тип сигналу(0 або 1). Якщо ця вершина якогось з інших типів, то далі іде одне або два числа, які позначають дітей, від яких бере сигнал дана вершина.

Для кращого розуміння, дивіться вхідні дані.

**Вихідні дані:**

Виведіть рядок з символів "0" або "1" – відповідь на задачу для кожного входу, в порядку їх сортування в вхідних даних.

**Приклади:**

|  |  |
| --- | --- |
| 3  OR 2 3  IN 0  IN 1 | 10 |
| 20  OR 17 10  IN 0  IN 0  NOT 6  OR 18 14  IN 1  OR 16 3  XOR 5 4  IN 0  XOR 11 9  NOT 15  AND 20 19  IN 0  IN 1  IN 1  NOT 8  NOT 12  IN 1  AND 13 7  NOT 2 | 11111111 |

Зображення до першого прикладу (жовтий – 0, зелений – 1): початковий стан, змінений сигнал другого елемента, змінений сигнал третього елемента

