Дано дерево

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Дано бесконечное бинарное дерево. У дерева есть корень и бесконечное число вершин, у каждой вершины есть левый и правый сын, у всех вершин кроме корня есть отец.

Каждая вершина может быть покрашена в один из c цветов или быть бесцветной. Изначально все вершины бесцветные.

Вам необходимо обрабатывать два типа запросов:

1. `color(`u`, `x`)` Дана вершина u, покрасить вершину u в цвет x, а затем вызвать `color(`L`, `$(x+1) \bmod c$`)` для ее левого сына L и `color(`R`, `$(x-1+c) \bmod c$`)` для её правого сына R. Заметим, что эта операция перекрашивает все (бесконечное) множество вершин в поддереве вершины u. Здесь mod — операция взятия числа по модулю. Если вершина уже была покрашена, то её цвет меняется на новый.

2. Дана вершина, вывести её текущий цвет.

Формат входных данных

В первой строке вводятся два числа q, c — количество запросов и цветов, соответственно ($1 \leqslant q \leqslant 5 \cdot 10^5$, $1 \leqslant c \leqslant 10^9$). Затем следует q запросов, каждый из которых начинается с целого числа t_i — типа i-го запроса.

Если $t_i = 1$, то далее в строке даётся целое число x ($0 \leqslant x \leqslant c-1$) цвет, в который надо покрасить вершину запроса u. В следующей строке описан путь до вершины u в виде непустой строки s_i, состоящей из символов «`L`» и «`R`». Данная строка задаёт путь от корня дерева до вершины u, где «`L`» обозначает переход к левому сыну, а «`R`» — к правому.

Если $t_i = 2$, то в следующей строке задаётся путь до вершины, цвет которой необходимо вывести, заданный аналогично предыдущему запросу.

Гарантируется, что сумма длин путей до всех вершин запросов не превосходит $5 \cdot 10^5$.

Формат выходных данных

Для каждого запроса второго типа в новой строке необходимо вывести ответ на него. Если вершина бесцветная, необходимо вывести число -1.

Пример

стандартный ввод	стандартный вывод
6 3 1 2 L 2 LL 1 0 LL 2 LLR 2 LR 2 R	0 2 1 -1