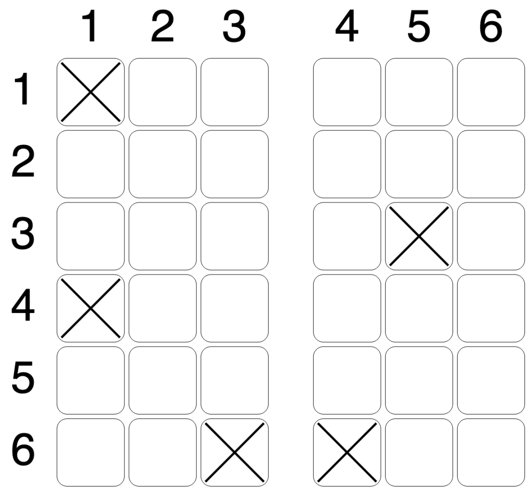
Задача 1. Посадка в самолет

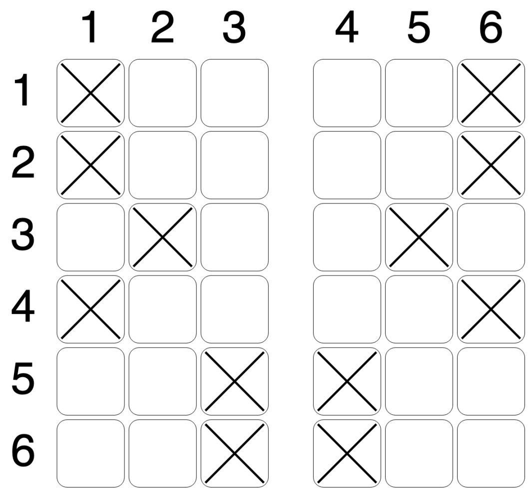
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В самолетах авиакомпании Битавиа кресла расположены в n рядов, при этом в каждом ряду по шесть мест, между третьим и четвертым местом находится проход. Некоторые пассажиры регистрируются заранее онлайн, другие пассажиры регистрируются на стойке регистрации в аэропорту.

При онлайн-регистрации пассажир может выбрать любое место и не может его затем менять. Например, при $n = 6$ рассадка в самолете после онлайн-регистрации может выглядеть так (крестиками отмечены занятые места)



На стойку регистрации придут m пассажиров. По правилам Битавиа нужно рассадить их в самолете таким образом, чтобы итоговая рассадка в самолете была симметрична относительно прохода. То есть, если в некотором ряду на первом кресле сидит пассажир, то в том же ряду на шестом кресле тоже должен сидеть пассажир. То же самое справедливо для второго и пятого, третьего и четвертого кресел, соответственно. При этом пересаживать пассажиров, прошедших онлайн-регистрацию нельзя. В исходную рассадку, показанную на рисунке выше, можно добавить семь пассажиров, удовлетворив условие симметрии например следующим образом:



Вам дана рассадка пассажиров после онлайн-регистрации. Требуется рассадить m пассажиров так, чтобы итоговая рассадка в самолете была симметрична относительно прохода, или определить, что это невозможно.

Формат входных данных

В первой строке содержатся два целых числа n и m — количество рядов в самолете и количество пассажиров, которые придут на стойку регистрации ($1 \leqslant n \leqslant 1000$, $0 \leqslant m \leqslant 6000$).

В следующих n строках задана изначальная рассадка в самолете после онлайн-регистрации. В каждой строке содержится по шесть символов, при этом i-й символ j-й строки равен «X» (заглавная английская X), если i-е место в j-м ряду уже занято и «.» (точка) иначе.

Формат выходных данных

Если искомой рассадки не существует, выведите «Impossible».

Иначе выведите n строк по шесть символов — итоговую рассадку в самолете. При этом i-й символ j-й строки должен быть равен «X», если место занято, и «.», если свободно. Если существует несколько решений, разрешается вывести любое.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	15	$m = 0$		первая ошибка
2	16	Изначально в самолете все места свободны		первая ошибка
3	17	$m = 1$		первая ошибка
4	18	Изначально в самолете занято ровно одно место		первая ошибка
5	34	нет	1–4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
1 0 X.XX.X	X.XX.X
2 1 X.XX.X ..X...	X.XX.X ..XX..
3 2 X.XX.X ...... X..X.X	Impossible
1 103 .X.XXX	Impossible
6 7 X..... ....X. X..... ....XX..	X....X X....X .X..X. X....X ..XX.. ..XX..

Замечание

Выше приведены пять примеров входных данных.

1) В первом примере $m = 0$, а рассадка в самолете симметрична, поэтому итоговая рассадка совпадает с исходной.

2) Во втором примере есть только один способ рассадить пассажиров симметрично.

3) В третьем примере существовало бы решение, при $m = 1$, но при $m = 2$ не существует способа рассадить всех пассажиров симметрично.

4) В четвертом примере требуется рассадить больше пассажиров чем свободных мест в самолете.

5) Пятый примере соответствует ситуации, рассмотренной на рисунках в тексте условия. В этом примере существует несколько решений, приведено одно из них.

Задача 2. Битоническая последовательность

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Последовательность $[b_1, b_2, \ldots, b_k]$ называется битонической, если выполнены неравенства $b_1 < b_2 < \ldots < b_i > \ldots > b_k$ для некоторого $1 \leqslant i \leqslant k$.

Например, последовательности $[1]$, $[1, 2, 3, 2]$, $[1, 4, 10]$, $[3, 2]$ являются битоническими, а последовательности $[1, 1]$, $[2, 1, 3]$ — нет

Задана последовательность $[a_1, a_2, \ldots, a_n]$. Требуется количество пар (l, r) таких, что $1 \leqslant l \leqslant r \leqslant n$ и последовательность $[a_l, a_{l+1}, \ldots, a_r]$ является битонической

Формат входных данных

Первая строка ввода содержит число n $(1 \leqslant n \leqslant 300\,000)$

Вторая строка ввода содержит n целых чисел: $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant n)$.

Формат выходных данных

Выведите одно число — количество пар (l, r), таких, что $1 \leqslant l \leqslant r \leqslant n$ и последовательность $[a_l, a_{l+1}, \ldots, a_r]$ является битонической.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	27	$n \leqslant 500$		первая ошибка
2	14	$n \leqslant 5000$	1	первая ошибка
3	20	все числа a_i различны		первая ошибка
4	39	—	1–3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 1 1 2 3 1	11
3 1 1 1	3

Замечание

В первом примере подходят следующие пары:

- $(1, 1)$, последовательность $[1]$
- $(2, 2)$, последовательность $[1]$
- $(2, 3)$, последовательность $[1, 2]$
- $(2, 4)$, последовательность $[1, 2, 3]$
- $(2, 5)$, последовательность $[1, 2, 3, 1]$
- $(3, 3)$, последовательность $[2]$
- $(3, 4)$, последовательность $[2, 3]$
- $(3, 5)$, последовательность $[2, 3, 1]$
- $(4, 4)$, последовательность $[3]$
- $(4, 5)$, последовательность $[3, 1]$
- $(5, 5)$, последовательность $[1]$

Задача 3. Игра с таблицей

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дана таблица A из h строк и w столбцов, в каждой ячейке которой записано целое число. Строки пронумерованы от 1 до h сверху вниз, столбцы пронумерованы от 1 до w слева направо.

Разрешается применять к этой таблице следующие операции:

- выбрать столбец таблицы и удалить его (столбцы слева и справа от него становятся соседними);
- выбрать строку таблицы и удалить ее (строки сверху и снизу от нее становятся соседними).

Эти операции разрешается применить произвольное число раз в любом порядке.

Определите, возможно ли при помощи этих операций получить из исходной таблицу с суммой чисел, равной заданному числу s, и если да, то какие операции и в каком порядке необходимо применить.

Формат входных данных

Первая строка ввода содержит числа h и w — размеры таблицы ($1 \leqslant h, w \leqslant 15$).

Каждая из следующих h строк содержит по w целых чисел — таблицу A ($0 \leqslant A_{i,j} \leqslant 10^9$)

В последней строке ввода находится число s — необходимая сумма ($1 \leqslant s \leqslant 10^{18}$)

Формат выходных данных

Если получить таблицу с суммой чисел s из исходной невозможно, выведите строку «`NO`»

Иначе:

- В первой строке выведите строку «`YES`».
- Во второй строке выведите единственное число k — количество операций с таблицей, которые необходимо применить, чтобы получить из неё таблицу с суммой чисел s.
- В каждой из следующих k строк выведите по два целых числа t_j, i_j, где $t_j = 1$, если очередная операция производится со строкой, и $t_j = 2$, если она производится со столбцом таблицы. Число i_j должно быть равно номеру строки или столбца, соответственно, в *исходной* нумерации, с которой эта операция производится.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	17	$h = 1$		первая ошибка
2	6	сумма чисел в i-й строке не превосходит i		первая ошибка
3	10	$h \leqslant 3$	1	первая ошибка
4	13	$h, w \leqslant 10$		первая ошибка
5	13	$h, w \leqslant 12$	4	первая ошибка
6	12	$a_{i,j} \leqslant 6$		первая ошибка
7	29		1–6	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 3 2 3 1 3 1 2 8	YES 2 1 3 2 3
2 3 2 2 2 2 2 2 5	NO
5 5 1 2 1 4 5 2 5 4 1 2 4 2 4 3 1 5 5 3 2 4 1 2 4 5 2 34	YES 3 1 4 1 5 2 1

Замечание

В первом примере изначально дана следующая таблица:

1	2	3
2	3	1
3	1	2

Удалив третьи строку и столбец получим таблицу с суммой чисел 8:

1	2	3
2	3	1
3	1	2

$\rightarrow$

1	2	3
2	3	1

$\rightarrow$

1	2
2	3

Во втором примере можно показать, что разрешенными операциями невозможно получить таблицу с суммой чисел 5 из исходной.

В третьем примере изначально дана таблица:

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4
1	2	4	5	2

Удалив последние две строки и первый столбец, получим таблицу с суммой чисел 34:

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4
1	2	4	5	2

$\rightarrow$

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1
5	5	3	2	4

$\rightarrow$

1	2	1	4	5
2	5	4	1	2
4	2	4	3	1

$\rightarrow$

2	1	4	5
5	4	1	2
2	4	3	1

Задача 4. Выбор столицы

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дано неориентированное дерево — связный граф из n вершин без циклов, и число k. Зафиксируем некоторую вершину s дерева и назовем ее столицей.

Ориентируем ребра дерева в направлении от столицы. Иными словами, ориентируем ребро (u, v) в направлении $u \to v$, если при подвешивании дерева за вершину s вершина u является родителем вершины v. Заметим, что при таком ориентировании ребер каждая вершина достижима из столицы

Определим расстояние до вершины v графа как минимальное количество ребер на пути из s в v. Назовем *доступностью* вершины s максимальное из расстояний до всех вершин

Разрешается добавить в дерево не более k дополнительных ориентированных ребер

Для каждой вершины s дерева определите, какой минимальной *доступности* можно достичь, если выбрать вершину s в качестве столицы.

Обратите внимание, что в некоторых подзадачах требуется вывести ответ только для первой вершины.

Формат входных данных

Первая строка содержит три целых числа n, k и t ($2 \leqslant n \leqslant 2 \cdot 10^5$, $1 \leqslant k \leqslant n-1$, $n \cdot k \leqslant 2 \cdot 10^5$, $0 \leqslant t \leqslant 1$) — количество вершин дерева, ограничение на максимальное количество добавленных ребер и число t, равное 0, если нужно вывести ответ только для вершины с номером 1, и равное 1 иначе.

Каждая из следующих $n-1$ строк содержит два целых числа u_i, v_i ($1 \leqslant u_i, v_i \leqslant n$) — ребра дерева.

Гарантируется, что заданные ребра образуют дерево.

Формат выходных данных

В случае, если $t = 0$, выведите единственное целое число: минимальную *доступность*, которую можно достичь, выбрав вершину с номером 1 в качестве столицы, и добавив не более k дополнительных ориентированных ребер.

В случае, если $t = 1$, выведите n чисел: i-е число равняется минимальной *доступности*, которую можно достичь, выбрав вершину i в качестве столицы, и добавив не более k дополнительных ориентированных ребер.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	5	$u_i = i, v_i = i+1, t = 0$		первая ошибка
2	5	$k = 1, n \leqslant 2000, t = 0$		первая ошибка
3	10	$k = 1, t = 0$	2	первая ошибка
4	5	$u_i = i, v_i = i+1$	1	первая ошибка
5	5	$n \leqslant 16$		первая ошибка
6	10	$n \leqslant 50$	5	первая ошибка
7	10	$n \leqslant 400$	5, 6	первая ошибка
8	10	$n \leqslant 2000$	5, 6, 7	первая ошибка
9	25	$n \cdot k \leqslant 50000$	2, 5, 6, 7, 8	первая ошибка
10	15	нет	1–9	первая ошибка

Примеры

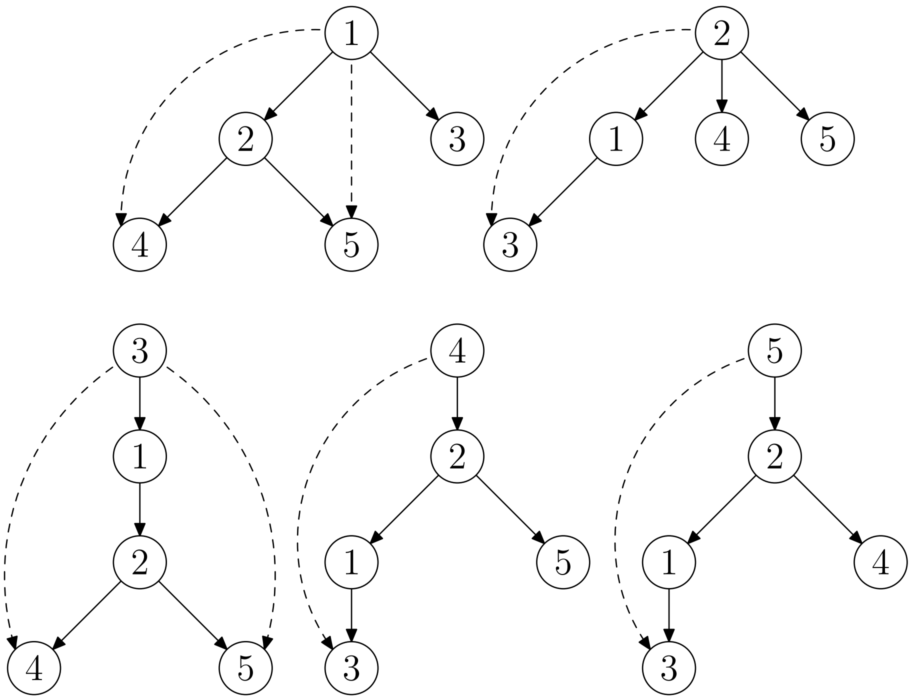
стандартный ввод	стандартный вывод

`5 2 1`
`1 2`
`1 3`
`2 4`
`2 5`
	`1 1 2 2 2`

`3 1 0`
`1 2`
`2 3`
	`1`

Замечание

На рисунке приведены иллюстрации к первому примеру. Пунктирными линиями обозначены добавленные ребра. Для вершин 1 и 2 минимальная *доступность* равняется 1, а для вершин 3, 4 и 5 минимальная *доступность* равняется 2.



Задача 5. Разбиение массива

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан массив $A = [a_1, a_2, \ldots, a_n]$, содержащий n натуральных чисел.

Требуется раскрасить элементы массива в два цвета таким образом, чтобы не существовало двух элементов x и y одного цвета, таких, что x нацело делился на y и выполнялось равенство $\frac{x}{y} = p$, где p — *простое* число. Гарантируется, что такая раскраска существует

Напомним, что целое число $p > 1$ называется простым, если оно имеет ровно два делителя: 1 и p

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 100\,000$) — количество элементов в массиве.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^6$) — элементы массива.

Формат выходных данных

Выведите описание разбиения массива на два множества в следующем формате

Выведите n целых чисел, i-е из которых равняется 1, если элемент a_i надо раскрасить в первый цвет, и 2, если элемент a_i надо раскрасить во второй цвет.

Если существует несколько подходящих раскрасок, вы можете вывести любую из них.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	9	$a_i \leqslant 2$ для всех i		первая ошибка
2	19	Гарантируется, что все a_i являются степенями некоторого простого числа p		первая ошибка
3	12	$a_i \leqslant 3$ для всех i	1	первая ошибка
4	13	$a_i \leqslant 4$ для всех i	1, 3	первая ошибка
5	21	$n \leqslant 10$		первая ошибка
6	26	нет	1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	2 1 1 2
1 20	1

Замечание

В первом примере есть два элемента первого цвета: 2 и 3, и два элемента второго цвета: 1 и 4. Элементы первого цвета не делятся нацело друг на друга. 4 нацело делится на 1, но их отношение не является простым числом.

Задача 6. Бактерии

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В биологической лаборатории проводят эксперимент. В начале у ученых есть n *замороженных* бактерий, пронумерованных от 1 до n.

Согласно плану эксперимента замороженная бактерия с номером i попадёт в чашку Петри через a_i секунд после начала эксперимента. Если таких бактерий несколько, они все попадают туда одновременно

Как только замороженная бактерия оказывается в чашке Петри, она размораживается и начинает *созревать*. Созревание бактерии с номером i занимает t_i секунд. Как только бактерия созрела, она начинает размножаться: немедлено превращается в две *созревшие* бактерии, и затем каждая созревшая бактерия в конце каждой секунды снова делится на две созревшие бактерии

Размером колонии называется общее количество бактерий в чашке Петри. Цель эксперимента — определить, через сколько секунд размер колонии будет в точности равен m

Помогите ученым определить искомое число секунд или выясните, что размер колонии никогда не будет в точности равен m.

Формат входных данных

В первой строке даны целые числа n, m ($1 \leqslant n \leqslant 2 \cdot 10^5$, $1 \leqslant m \leqslant 10^9$) — количество замороженных бактерий и желаемый размер колонии.

Во второй строке даны n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — времена перемещения замороженных бактерий в чашку Петри.

В третьей строке даны n целых чисел $t_1, t_2, \ldots, t_n$ ($1 \leqslant t_i \leqslant 10^9$) — продолжительность созревания замороженных бактерий.

Формат выходных данных

Если размер колонии никогда не будет равен m, выведите -1.

В противном случае выведите число секунд после начала эксперимента, через которое размер колонии будет в точности равен m.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	13	$m \leqslant n$, $a_i \leqslant 10^5$, $t_i = 10^9$		первая ошибка
2	14	$a_i = i$, t_i равны		первая ошибка
3	17	$n, a_i, t_i \leqslant 3000$		первая ошибка
4	23	a_i равны 1		первая ошибка
5	33	—	1–4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 11 3 5 1 10 2 9 2 13	5
13 124 5 6 8 8 1 6 4 6 4 7 10 3 9 5 2 10 5 2 1 1 4 8 3 4 1 9	8

Замечание

Рассмотрим, как развивается эксперимент в первом примере.

Время	**Бактерия 1**	**Бактерия 2**	**Бактерия 3**	**Бактерия 4**	**Всего**
0	заморожена	заморожена	заморожена	заморожена	0
1	заморожена	заморожена	в чашке Петри, созревает	заморожена	1
2	заморожена	заморожена	в чашке Петри, созревает	заморожена	1
3	в чашке Петри, созревает	заморожена	в чашке Петри, созрела, 2 бактерии	заморожена	3
4	в чашке Петри, созревает	заморожена	в чашке Петри, созрела, 4 бактерии	заморожена	5
5	в чашке Петри, созрела, 2 бактерии	в чашке Петри, созревает	в чашке Петри, созрела, 8 бактерий	заморожена	11

Задача 7. Разбиение на тройки

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

На день рождения Маше как обычно подарили массив a из n натуральных чисел, в котором каждое число находится в пределах от 1 до m включительно. Маша очень любит число три, поэтому длина массива делится на три.

Маша решила объединять числа в *тройки*: каждая тройка чисел должна состоять или из трех одинаковых чисел, или из трех последовательных чисел. Другими словами, каждая тройка имеет или вид (x, x, x), или $(x, x+1, x+2)$, где x — какое-то натуральное число

Маша хочет поиграть с подаренным массивом, и ее интересует количество способов разбить числа этого массива на такие тройки. Два способа разбиения считаются различными, если нельзя установить взаимно-однозначное соответствие между тройками первого разбиения и тройками второго разбиения, что числа внутри соответствующих троек равны. Так как количество разбиений может быть большим, Маше достаточно знать его остаток по модулю $10^9 + 7$

Помогите Маше посчитать количество способов разбить числа подаренного ей массива на тройки по модулю $10^9 + 7$

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($1 \leqslant n \leqslant 5000$, $1 \leqslant m \leqslant 5000$, $n = 3 \cdot k$ для какого-то натурального k).

Вторая строка содержит n целых чисел a_i — числа массива ($1 \leqslant a_i \leqslant m$).

Формат выходных данных

В единственной строке одно число — количество способов разбить числа массива на тройки по модулю $10^9 + 7$.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$m \leqslant 3$		первая ошибка
2	8	$m \leqslant 4$	1	первая ошибка
3	10	каждое число от 1 до m встречается не более двух раз		первая ошибка
4	12	массив a не содержит чисел, которые делятся на 4	1	первая ошибка
5	29	$n \leqslant 500$, $m \leqslant 500$		первая ошибка
6	31	—	1, 2, 3, 4, 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
9 4 3 4 2 4 4 2 3 3 2	2
6 3 1 2 3 1 2 1	0

Замечание

В первом примере числа можно разбить на тройки двумя способами: $\{(2, 2, 2), (3, 3, 3), (4, 4, 4)\}$ и $\{(2, 3, 4), (2, 3, 4), (2, 3, 4)\}$.

Задача 8. Обходы бинарного дерева

Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Бинарное дерево — это набор вершин, у каждой из которых может быть левый и правый ребёнок. Одна из вершин является корнем дерева, она не является ребёнком какой-то другой. Начав в корне и каждый раз переходя в одного из детей, можно дойти до любой вершины. Множество вершин, до которых можно дойти из заданной, называется её поддеревом.

У бинарного дерева есть три основных обхода: прямой (*pre-order*), центрированный (*in-order*) и обратный (*post-order*).

Прямой обход дерева — это порядок его вершин, полученный следующим рекурсивным алгоритмом:

1. Добавить корень дерева в обход.
2. Если у корня есть левый ребёнок, выписать прямой обход его поддерева.
3. Если у корня есть правый ребёнок, выписать прямой обход его поддерева.

В центрированном обходе корень дерева выписывается между обходами поддеревьев его детей, в обратном — после обходов поддеревьев его детей.

Обобщим эти три варианта обхода: пусть в каждой вершине записано целое число x от -1 до 1, обозначающее, в какой момент мы выписываем эту вершину, а именно:

- $x = -1$: до обходов поддеревьев её детей;
- $x = 0$: между обходами поддеревьев её детей;
- $x = 1$: после обходов поддеревьев её детей.

Таким образом, если во всех вершинах записано -1, обход является прямым, если 0 — центрированным, если 1 — обратным.

Рассмотрим дерево с n вершинами, пронумерованных от 1 до n. Корень дерева — вершина 1. Изначально во всех вершинах записано число -1.

В рамках исследования необходимо обработать q запросов одного из следующих типов:

1. Поменять числа в вершинах $l, l+1, \ldots, r$ на x (x равен -1, 0 или 1).
2. Сообщить, на какой позиции в текущем обходе будет стоять вершина i.

Необходимо вывести ответы на все запросы второго типа.

Формат входных данных

В первой строке входных данных даны два целых числа n и q ($1 \leqslant n, q \leqslant 100\,000$).

В следующих n строках даны по два целых числа L_i и R_i ($0 \leqslant L_i, R_i \leqslant n$) — номер левого и правого ребёнка вершины i соответственно, либо 0, если соответствующий ребёнок отсутствует

Гарантируется, что L_i и R_i задают корректное бинарное дерево

В следующих q строках даны запросы. Первое число в строке t ($t \in \{1, 2\}$) — тип запроса

В случае запроса первого типа далее даны целые числа l, r и x ($1 \leqslant l \leqslant r \leqslant n$, x равен -1, 0 или 1) — границы отрезка вершин, в которых меняются числа, и новое значение.

В случае запроса второго типа далее дано число i ($1 \leqslant i \leqslant n$) — номер вершины, позицию которой в обходе необходимо вывести.

Формат выходных данных

На каждый запрос второго типа выведите единственное число от 1 до n — позицию соответствующей вершины в обходе.

Система оценки

Пусть q_1 — количество запросов первого типа.

Подзадача	Баллы	Дополнительные ограничения	Необх. подзадачи	Информация о проверке
1	10	$n, q \leqslant 5000$		первая ошибка
2	5	$q_1 \leqslant 10$		первая ошибка
3	10	все запросы первого типа идут до всех запросов второго типа		первая ошибка
4	10	все листья (вершины без детей) находятся на одном расстоянии от корня, нет вершин с ровно одним ребёнком		первая ошибка
5	10	$l = r$ для всех запросов первого типа		первая ошибка
6	20	$x \in \{-1, 1\}$ для всех запросов первого типа, у каждой вершины не более одного ребёнка		первая ошибка
7	10	$x \in \{-1, 1\}$ для всех запросов первого типа	6	первая ошибка
8	10	у каждой вершины не более одного ребёнка	6	первая ошибка
9	15	нет	1–8	первая ошибка

Пример

стандартный ввод	стандартный вывод
5 5 3 4 0 0 5 2 0 0 0 0 2 2 1 1 3 1 2 5 1 3 3 0 2 3	4 1 2

Замечание

В примере обход меняется следующим образом:

- $[1, 3, 5, 2, 4]$
- $[5, 2, 3, 4, 1]$
- $[5, 3, 2, 4, 1]$