Задача A. Двоичное дерево поиска

Имя входного файла:	`bst.in`
Имя выходного файла:	`bst.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Реализуйте сбалансированное двоичное дерево поиска.

Формат входных данных

Входной файл содержит описание операций с деревом, их количество не превышает 100000. В каждой строке находится одна из следующих операций:

- `insert` x — добавить в дерево ключ x. Если ключ x в дереве уже есть, то ничего делать не надо.
- `delete` x — удалить из дерева ключ x. Если ключа x в дереве нет, то ничего делать не надо.
- `exists` x — если ключ x есть в дереве, выведите «`true`», иначе «`false`»
- `next` x — выведите минимальный элемент в дереве, строго больший x, или «`none`», если такого нет.
- `prev` x — выведите максимальный элемент в дереве, строго меньший x, или «`none`», если такого нет.

Все числа во входном файле целые и по модулю не превышают 10^9.

Формат выходных данных

Выведите последовательно результат выполнения всех операций `exists`, `next`, `prev`. Следуйте формату выходного файла из примера.

Примеры

bst.in	bst.out
insert 2 insert 5 insert 3 exists 2 exists 4 next 4 prev 4 delete 5 next 4 prev 4	true false 5 3 none 3

Задача B. Асхат и дерево

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Дано двоичное дерево поиска размера n, в каждой вершине есть значение от 1 до n. Петух по имени Асхат каждый день нумерует вершины числами от 1 до n, даёт вам номер корня r и просит вас найти номер вершины со значением x.

Вы можете совершать запросы — по номеру вершины узнать значение в ней и номера её детей. Пусть максимальное расстояние от корня до вершины в i-й день равно d. Тогда в i-й день вы можете совершить не более, чем d запросов. За все дни вы можете совершить не более, чем 70000 запросов.

Также вы можете менять детей у каждой вершины. Пусть в i-й день длина пути от корня до вершины с номером x равна s. Тогда вам в i-й день разрешено сделать не более, чем s запросов вида «установить у вершины новых детей». За все дни вы можете совершить не более, чем 70000 запросов этого типа.

Протокол взаимодействия

В первой строке ввода даны числа n и q ($1 \leqslant n \leqslant 2000$, $1 \leqslant q \leqslant 2000$) — размер дерева и число запросов, соответственно.

Для каждого запроса даны числа r и x ($1 \leqslant r, x \leqslant n$) — номер корня дерева и значение, вершину с которым требуется найти. Вы не сможете считать эти числа для следующего запроса, пока не дадите ответ на текущий.

Чтобы обратиться к вершине с номером i выведите «`val i`» в отдельной строке. В ответ даются три числа val, L и R ($1 \leqslant val \leqslant n$, $0 \leqslant L, R \leqslant n$) — значение в этой вершине и номера левого и правого ребёнка, соответственно. В случае, если у вершины нет левого или правого ребёнка, $L = 0$ или $R = 0$, соответственно.

Чтобы поменять детей вершины с номером i на вершины с номером L и R выведите «`change i L R`» в отдельной строке. Чтобы у вершины с номером i не было левого или правого ребёнка, выведите 0 вместо L или R, соответственно. После выполнения этого запроса граф может перестать быть двоичным деревом поиска.

Если искомое значение находится в вершине с номером i и вы совершили все нужные изменения, выведите «`confirm i`» в отдельной строке. После этого вершины перенумеруются, а на ввод будет дан новый запрос. Если на момент выполнения этого запроса граф не является двоичным деревом поиска, вы получите вердикт «Wrong answer».

Задача C. Своппер

Имя входного файла:	`swapper.in`
Имя выходного файла:	`swapper.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Современные компьютеры зацикливаются
в десятки раз эффективнее человека

Рекламный проспект OS Vista-N

Перед возвращением в штаб-квартиру корпорации Аазу и Скиву пришлось заполнить на местной таможне декларацию о доходах за время визита. Получилась довольно внушительная последовательность чисел. Обработка этой последовательности заняла весьма долгое время.

— Своппер кривой, — со знанием дела сказал таможенник.

— А что такое своппер? — спросил любопытный Скив.

Ааз объяснил, что своппер — это структура данных, которая умеет делать следующее.

- Взять отрезок чётной длины от x до y и поменять местами число x с $x+1$, $x+2$ с $x+3$, и т.д.
- Посчитать сумму чисел на произвольном отрезке от a до b.

Учитывая, что обсчёт может затянуться надолго, корпорация «МИФ» попросила Вас решить проблему со своппером и промоделировать ЭТО эффективно.

Формат входных данных

Во входном файле заданы один или несколько тестов. В первой строке каждого теста записаны число N — длина последовательности и число M — число операций ($1 \leqslant N, M \leqslant 100\,000$). Во второй строке теста содержится N целых чисел, не превосходящих 10^6 по модулю — сама последовательность. Далее следуют M строк — запросы в формате `1` x_i y_i — запрос первого типа, и `2` a_i b_i — запрос второго типа. Сумма всех N и M по всему файлу не превосходит 200 000. Файл завершается строкой из двух нулей. Гарантируется, что $x_i < y_i$, а $a_i \leqslant b_i$.

Формат выходных данных

Для каждого теста выведите ответы на запросы второго типа, как показано в примере. Разделяйте ответы на тесты пустой строкой.

Примеры

swapper.in	swapper.out
5 5 1 2 3 4 5 1 2 5 2 2 4 1 1 4 2 1 3 2 4 4 0 0	Swapper 1: 10 9 2

Задача D. Переворот

Имя входного файла:	`reverse.in`
Имя выходного файла:	`reverse.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дан массив. Надо научиться обрабатывать два типа запросов.

- `1 L R` - перевернуть отрезок [L, R]
- `2 L R` - найти минимум на отрезке [L, R]

Формат входных данных

Первая строка файла содержит два числа n, m. ($1 \leqslant n, m \leqslant 10^5$) Во второй строке находится n чисел a_i ($1 \leqslant a_i \leqslant 10^9$)- исходный массив. Остальные m строк содержат запросы, в формате описанном в условии. Для чисел L,R выполняется ограничение ($1 \leqslant L \leqslant R \leqslant n$).

Формат выходных данных

На каждый запрос типа 2, во входной файл выведите ответ на него, в отдельной строке.

Примеры

reverse.in	reverse.out
10 7 5 3 2 3 12 6 7 5 10 12 2 4 9 1 4 6 2 1 8 1 1 8 1 8 9 2 1 7 2 3 6	3 2 2 2