

Задача А. Ценные бумаги

Максимальное время работы на одном тесте:	4 секунды
Максимальный объем используемой памяти:	64 мегабайта
Название входного файла:	bonds.in
Название выходного файла:	bonds.out

Ценные бумаги на фондовом рынке характеризуются множеством параметров. У них есть цена и ликвидность, также оценивать динамичность изменения цены, среднюю прибыльность, потенциал роста прибыльности и др. показатели. Аналитики трейдовой компании "WebMarket" ввели специальный показатель *надежности ценной бумаги* и научились эффективно его оценивать. Большое значение этого показателя соответствует малому риску покупки ценной бумаги. Но с ростом надежности обычно падает среднее оцениваемое значение прибыльности.

Для своих клиентов, играющих на рынке ценных бумаг, компания "WebMarket" решила открыть значения этого показателя и более того, автоматизировать покупку ценных бумаг с заданным порядковым номером по значению надежности. Аналитики проанализировали идею, и решили, что наличие такого функционала будет способствовать привлечению новых клиентов на рынок "WebMarket", повышению объемов сделок, а значит, и повышению прибылей "WebMarket". Важно также отметить, что торговля на базе этого показателя может позитивно сказаться на российском фондовом рынке и сделать его более здоровым. Алгоритмы оценки надежности уже написаны, средства выделены, необходимая реклама проведена. Осталось только написать сам код.

Ценные бумаги в базе данных имеют три атрибута:

- code — непустая строка латинских символов и цифр длины 30 или меньше
- id — целочисленный идентификатор (начиная с 0)
- reliability — целое число из диапазона $[-2^{31}, 2^{31})$.

Каждой новой ЦБ выдается следующий по порядку id и значение её надёжности устанавливается в 0. Если ценная бумага отзывается с рынка, её id для новых бумаг не используется.

База данных получает запросы, которые позволяют вводить новые ЦБ на рынок, получать текущую информацию о ЦБ, отзываться ЦБ с рынка, менять значение надежности у ЦБ и находить ЦБ, которая стоит на n-м месте, если упорядочить ЦБ по убыванию надежности, а при одинаковых значениях по возрастанию идентификатора.

При добавлении ЦБ с кодом, который раньше встречался, но соответствующая ЦБ была отозвана с рынка, ей назначается новый идентификатор.

Входные и выходные данные

Первая строка входа содержит число запросов n . ($1 \leq n \leq 100000$). Затем идут n строк, каждая из которых содержит один запрос. Запросы бывают 4 типов:

- ISSUE code: добавить новую ЦБ code в базу данных, или вывести информацию о ЦБ, если она уже есть в базе
- если ЦБ code существует, то вывести
- EXISTS id reliability
- если ЦБ code не существует, то вывести
- CREATED id 0
- DELETE code: удалить ЦБ code из базы
- если ЦБ code существует, то вывести
- OK id reliability
- если ЦБ code не существует, то вывести
- BAD REQUEST
- RELIABILITY code reliability: увеличить количество очков ЦБ, гарантируется, что значение останется в диапазоне $[-2^{31}, 2^{31})$.
- если ЦБ code существует, то вывести
- OK id new_reliability
- если ЦБ code не существует, то вывести
- BAD REQUEST
- FIND n: найти n-ю ЦБ (начиная с 0)
- если база не пуста, найти ЦБ, которая имела бы порядковый номер n (нумерация начинается с 0), если все ЦБ упорядочить сначала по убыванию баллов, а группы ЦБ с одинаковыми баллами

упорядочить по времени их добавления (то есть по возрастанию id); а если n больше, чем количество ЦБ в базе, то найти последнюю ЦБ в этом порядке; для обоих случаев вывести OK code id reliability

- если база пуста, то вывести
- EMPTY

Таким образом, на каждый запрос на входе нужно вывести одну строку с результатом.

Пример

Входные данные	Выходные данные
17	CREATED 0 0
ISSUE aaa	OK aaa 0 0
FIND 10	CREATED 1 0
ISSUE bbb	CREATED 2 0
ISSUE ccc	OK 0 10
RELIABILITY aaa 10	OK 1 30
RELIABILITY bbb 30	OK 2 20
RELIABILITY ccc 20	BAD REQUEST
RELIABILITY xxx 20	OK ccc 2 20
FIND 1	OK aaa 0 10
FIND 2	OK bbb 1 30
FIND 0	CREATED 3 0
ISSUE eee	CREATED 4 0
ISSUE fff	OK eee 3 0
FIND 3	OK fff 4 0
FIND 111	OK 1 30
DELETE bbb	OK ccc 2 20
FIND 0	

Задача В. Гаджеты

Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта
 Название входного файла: gadgets.in
 Название выходного файла: gadgets.out

Товарищ Иванов — очень богатый любитель гаджетов. Недавно он понял, что не может купить всех гаджетов, которые он хочет потому что их никто не производит. Поэтому он решил построить свою собственную Фабрику Гаджетов.

Фабрика Гаджетов будет построена на Сколковском шоссе. На этом шоссе сконцентрировано производство нанотехнологичной продукции, необходимой для производства гаджетов. Сколковское шоссе представляет собой прямую, а фабрики — точки на этой прямой (фабрики расположены вдоль дороги).

Для производства гаджета необходимо N нанотехнологичных деталей, M фабрик производит эти детали. Товарищ Иванов хочет минимизировать сумму квадратов расстояний до ближайших фабрик, производящих необходимые детали. Помогите ему и найдите все точки для которых эта сумма минимальна.

Формат входных данных

Первая строка входного файла содержит два натуральных числа N и M ($1 \leq n \leq 10000$; $N \leq M \leq 100000$)

Следующие M строк содержат пары чисел x_i и p_i , x_i — координата i -ой фабрики, а p_i — идентификатор детали, которую производит данная фабрика ($x_i \leq 100000$; $x_i \leq x_{i+1}$; $1 \leq p_i \leq N$).

Для каждой детали существует хотя бы одна фабрика, которая производит эту деталь.

Формат выходных данных

Первая строка выходного файла должна содержать целое число K — количество точек, где может быть расположена Фабрика Гаджетов.

Следующие K строк должны содержать координаты этих точек в возрастающем порядке. Значения должны выводиться с точность 10^{-6} .

Примеры

Входные данные	Выходные данные
3 5 -1 3 0 1 2 3 4 2 5 2	1 2.0
2 5 1 1 2 2 3 1 4 2 5 1	4 1.5 2.5 3.5 4.5