Задача A. Шарики

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

В стеклянной трубке, открытой с двух концов, находится несколько шариков. Петя ставит следующий опыт над шариками в трубке: в момент времени 0 некоторым шарикам он придаёт скорость 1 сантиметр в секунду в направлении левого конца трубки, а остальным — 2 сантиметра в секунду в направлении правого конца трубки. Шарики устроены так, что в момент столкновения шарика, катящегося влево со скоростью u сантиметров в секунду, и шарика, катящегося вправо со скоростью v сантиметров в секунду, скорости шариков меняются следующим образом: первый шарик получает скорость v сантиметров в секунду, и далее движется вправо, а второй шарик отскакивает влево со скоростью u сантиметров в секунду.

Петя утверждает, что по начальной конфигурации шариков и начальным скоростям может точно предсказать, сколько секунд потребуется, чтобы все шарики выкатились из трубки. Напишите программу, которая делает то же самое, чтобы проверить его предсказания.

Формат входного файла

В первой строке входного файла записаны три числа — количество шариков N, длина трубки L в сантиметрах и предсказанное Петей количество секунд P ($1 \leqslant N \leqslant 100\,000$, $1 \leqslant L \leqslant 10^9$, $0 \leqslant P \leqslant 10^9$). Далее следует N строк, по два числа X_i и V_i в каждой, задающих начальное удаление i-го шарика от левого конца трубки и его начальную скорость ($0 \leqslant X_i \leqslant L$, $V_i \in \{-1; 2\}$). Числа N, L и X_i — целые, число P имеет не больше шести знаков после десятичной точки. Размером шариков можно пренебречь.

Формат выходного файла

Выведите `YES`, если Петя прав, и предсказанное им время P в точности равно времени, когда последний шарик выкатится из трубки, и `NO`, если он ошибается.

Примеры

balls.in	balls.out
`1 3 1` `1 2`	`YES`
`1 2 1` `1 2`	`NO`
`1 2 0.5` `1 2`	`YES`

`2 1 0.012345`
`0 -1`
`1 2`
	`NO`

`3 10 4`
`2 2`
`3 -1`
`5 2`
	`YES`

Задача B. Весы

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

Ежи всегда носил с собой какую-то вещь.
Определённую...
Андрей Андрианов, “Ежи и Петруччо”

Роботу Кузе поручили взвесить вещь. Для этого ему были предоставлены двухчашечные весы и набор из N гирь. Ему подготовили рабочее место — положили вещь и некоторые гири на одну чашу весов, а остальные гири — на другую.

Да только вот беда: Кузя — весьма несовершенный робот. Начнём с того, что в силу конструктивных особенностей и расположения весов он не может просто снять гирю с весов. Всё, что может сделать робот — это переложить определённую гирю с одной чаши весов на другую. Хорошо хоть, что он различает гири и запоминает, где какая находится.

Другая проблема заключается в том, что Кузя не может по положению весов определить, какая чаша на данный момент легче, а какая — тяжелее. Лишь когда чаши весов приходят в равновесие, раздаётся мелодичный звон, который робот может уловить. Логично утверждать, что если бы весы изначально находились в положениии равновесия, то Кузе бы не поручили эту работу.

Наконец, хотя Кузя и различает все гири, он не имеет понятия о том, каков их вес. Он не может даже определить, верно ли, что два набора гирь имеют одинаковый вес — для этого ему пришлось бы убрать вещь с весов, а этого он сделать не может.

Понятно, что Кузе очень не нравится такая работа, и он хочет покончить с ней как можно скорее. Для этого ему, очевидно, необходимо будет перебрать все положения гирь, которые он может получить из исходного. Если при переборе всех положений весы ни разу не придут в равновесие, значит, Кузя не может точно взвесить данную ему вещь. В противном случае он остановится, как только услышит звон весов.

Кузя хочет минимизировать количество перемещений гирь. Помогите ему составить план того, когда какую гирю перемещать, чтобы в худшем случае, если точное взвешивание невозможно, перебрать все положения гирь и доказать это.

Формат входного файла

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 14$) — количество гирь. Во второй строке даны через пробел N чисел $A_1, A_2, \ldots, A_N$, каждое из которых равно 0 или 1. $A_i = 0$ означает, что гиря с номером i изначально находится на той же чаше весов, что и вещь, $A_i = 1$ — на противоположной.

Формат выходного файла

На первой строке выходного файла выведите одно число — количество перемещений гирь M ($0 \leqslant M \leqslant 1\,000\,000$) (помните, что M нужно минимизировать). На второй строке выведите M чисел B_j

($1 \leqslant B_j \leqslant N$) через пробел — какую гирю следует переложить на j-ом шаге с той чаши весов, на которой она сейчас находится, на противоположную.

Примеры

scales.in	scales.out
1 0	1 1
2 0 0	3 1 2 1
2 0 1	3 2 1 2

Задача C. Ленивый работник

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

Трактир «Не дай себе просохнуть!» взял на работу нового работника — кладовщика. У кладовщика есть две обязанности. Первая — принимать от грузчиков бочки с пивом и закатывать их в подвал. Вторая — когда в бочке у трактирщика не остаётся пива, выкатывать ему из подвала новую бочку, в которой пиво ещё имеется.

Хитрый работник скоро понял, что в разных бочках, попадающих к нему в подвал, содержится различное количество пива, и решил впредь всегда выкатывать трактирщику ту бочку, в которой пива менее всего, чтобы особо не напрягаться. Дальновидность не в характере лентяя-кладовщика, и то обстоятельство, что позднее ему придётся катать исключительно тяжёлые бочки, равно как и то, что лёгкие бочки нужно катать чаще, он проигнорировал.

При получении бочки от грузчиков кладовщик измеряет количество пива в ней. Количество пива обозначается целым числом от 0, когда бочка пуста, до 1000, когда вся бочка заполнена пивом. Известно, что грузчики не передают кладовщику пустых бочек.

Трактирщик узнал о том, какую хитрость придумал его новый ленивый работник, и теперь, зная количество пива во всех привозимых грузчиками бочках, хочет выяснить, сколько пива будет в каждой следующей бочке, которую кладовщик выкатит из подвала.

По последовательности поступлений бочек в подвал и выкатываний их трактирщику установите, сколько пива было в каждой выкаченной бочке.

Формат входного файла

В первой строке входного файла задано число N ($1 \leqslant N \leqslant 100\,000$). Следующие N строк содержат по одному числу A_i в каждой и описывают события в порядке, в котором они происходят. Если A_i больше нуля, то это означает поступление в подвал новой бочки, количество пива в которой равно A_i. Если же A_i равно нулю, то описываемое данной строкой событие — это выкатывание из подвала бочки, пива в которой осталось менее всего.

Изначально можно считать, что в подвале бочек с пивом нет. Гарантируется, что трактирщик не попадёт в ситуацию, когда на его запрос кладовщик не может выкатить новой бочки.

Формат выходного файла

На каждое выкатывание бочки из подвала выведите строку, состоящую из единственного числа B_j — сколько пива содержалось в выкаченной бочке.

Примеры

worker.in	worker.out
3 1 2 0	1
5 3 0 2 0 1	3 2
6 8 4 2 1 0 0	1 2
4 1 1 1 0	1
10 4 3 7 4 9 5 6 0 0 0	3 4 4