Задача A. Число

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

Дана последовательность из восьми цифр, каждая из которых равна 1, 2 или 3. Требуется выбрать из них четыре цифры так, чтобы число, полученное при выписывании этих цифр в том же порядке, в каком они были в последовательности, оказалось как можно меньше.

Формат входного файла

Первая строка входного файла содержит восемь цифр из набора $\{1, 2, 3\}$, записанных подряд без пробелов.

Формат выходного файла

В первой строке выходного файла выведите четыре цифры, образующие искомое минимальное число, также без пробелов.

Примеры

number.in	number.out
12333333	1233
32111111	1111
12131211	1111
33333323	3323
32333313	2313

Задача B. 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

Задача C. Весы-2

Имя входного файла:	`scales2.in`
Имя выходного файла:	`scales2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Роботу Кузе поручили взвесить ещё одну вещь. Как и ранее, ему были предоставлены двухчашечные весы и набор из N гирь. Ему подготовили рабочее место — положили вещь и все гири на одну чашу весов. Перемещать вещь запрещено.

Хотя Кузя не может определить ни вес каждой гири, ни какая чаша весов на данный момент перевесила, он всё же может различить гири и запоминает, где какая находится. Важное отличие от взвешивания, проделанного Кузей в прошлый раз, заключается в том, что робот теперь научился снимать гирю с любой чаши весов и ставить её рядом, а также брать снятую гирю и ставить на любую чашу весов. Кузя, правда, не очень-то обрадовался своим новым способностям — ведь количество возможных расстановок гирь из-за них сильно возросло.

Кузя знает, что, когда весы окажутся в положении равновесия, раздастся мелодичный звон, и его работа будет выполнена. Если же он переберёт все возможные расстановки гирь, но звона так и не услышит, это будет значить, что робот не может выполнить точное взвешивание с данными ему гирями, и работа также прекратится.

Кузе очень не нравится такая работа, и он хочет покончить с ней как можно скорее. Помогите ему составить план того, когда, куда и какую гирю перемещать, чтобы в худшем случае, если точное взвешивание невозможно, перебрать все положения гирь за минимальное количество перемещений. Предположения о том, что веса вещи и всех гирь положительны, делать не стоит: например, расстановку, в которой все гири сняты с весов, также требуется проверить.

Формат входного файла

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 12$) — количество гирь. Все гири изначально находятся на той же чаше весов, что и вещь.

Формат выходного файла

В первой строке выходного файла выведите одно число — количество перемещений гирь M ($0 \leqslant M \leqslant 1\,000\,000$); помните, что M нужно минимизировать. Далее выведите M строк вида A_i B_i. Здесь A_i ($1 \leqslant A_i \leqslant N$) задаёт номер перемещаемой гири, а B_i — куда её требуется перемещать. $B_i = 0$ означает снятие гири с весов, $B_i = 1$ — перемещение на чашу весов, на которой находится вещь, $B_i = 2$ — перемещение на чашу весов, на которой вещи нет. Запрещается перемещать гирю туда же, где она в данный момент находится; например, нельзя, чтобы B_1 было равно 1. Числа в строке разделяйте одним пробелом.

Если планов, минимизирующих M, несколько, разрешается выводить любой из них.

Примеры

scales2.in	scales2.out
1	2 1 0 1 2
2	8 1 0 2 0 1 1 1 2 2 1 2 2 1 0 1 1