Задача A. Правильная скобочная последовательность

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа дожна определить, является ли данная скобочная последовательность правильной.

Пустая последовательность является правильной. Если A — правильная, то последовательности (A), $[A]$, $\{A\}$ — правильные. Если A и B — правильные последовательности, то последовательность AB — правильная.

Формат входных данных

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

Формат выходных данных

Если данная последовательность правильная, то программа должна вывести строку *yes*, иначе строку *no*.

Примеры

стандартный ввод	стандартный вывод
`()[]`	`yes`
`([)]`	`no`

Задача B. Постфиксная запись

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $AB+$. Запись $BC + D*$ обозначает привычное нам $(B+C)*D$, а запись $ABC+D*+$ означает $A+(B+C)*D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Формат входных данных

В единственной строке записано выражение в постфиксной записи, содержащее цифры и операции $+$, $-$, $*$. Числа и операции разделяются пробелами. В конце строки может быть произвольное количество пробелов. Числа не превосходят 100 по модулю.

Формат выходных данных

Необходимо вывести значение записанного выражения.

Пример

стандартный ввод	стандартный вывод
8 9 + 1 7 - *	-102

Задача C. Шарики

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет «уничтожено». Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Формат входных данных

Сначала вводится количество шариков в цепочке $1 \leqslant n \leqslant 10^5$ и цвета шариков $0 \leqslant c_i \leqslant 9$.

Формат выходных данных

Требуется вывести количество шариков, которое будет «уничтожено».

Пример

стандартный ввод	стандартный вывод
5 1 3 3 3 2	3

Задача D. Очередь с минимумом

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вам дано q запросов трех типов:

- «1 x» — добавить число $0 \leqslant x \leqslant 10^{18}$ в конец очереди.
- «2» — вывести на экран и удалить первое число из очереди. Гарантируется, что очедерь в данный момент времени не пуста.
- «3 i» — вывести на экран i-е число очереди. Гарантируется, что $1 \leqslant i \leqslant len$, где len — это размер очереди в данный момент времени.
- «4» — вывести на экран минимальное число очереди. Гарантируется, что очедерь в данный момент времени не пуста.

Формат входных данных

В первой строке вводится одно число $1 \leqslant q \leqslant 10^5$ — количество запросов.
В последующие q строках вводятся запросы в выше описанном формате.

Формат выходных данных

Для каждого запроса второго, третьего и четвертого типа выведите по одному числу.

Пример

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

Задача E. Сумма Элементов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

У вас есть изначально пустой массив. Вам нужно обрабатывать запросы двух видов:

- Вид 1 x c: вам нужно добавить в конец массива c элементов, каждый из которых равен x
- Вид 2 c: вам нужно найти сумму первых c элементов массива, а затем удалить их. Гарантируется, что во время такого запроса в массиве находится хотя бы c элементов.

Формат входных данных

В первой строке вводится одно число Q ($1 \leqslant Q \leqslant 2 \cdot 10^5$) — число запросов.

Затем в следующих Q строках. Для запросов первого вида выполняется $0 \leqslant x \leqslant 10^9, 1 \leqslant c \leqslant 10^9$, а для запросов второго вида — $1 \leqslant c \leqslant 10^9$.

Все числа во входных данных — целые.

Формат выходных данных

Выведите ответы на все запросы второго типа в том порядке, в котором они идут во входных данных.

Примеры

стандартный ввод	стандартный вывод
4 1 2 2 2 1 1 4 5 2 3	2 10
2 1 1000000000 1000000000 2 1000000000	1000000000000000000

Замечание

Покажем, как будет меняться массив в первом примере:

1. Первая операция — добавить два числа 2 в массив. После этой операции массив будет равен $[2, 2]$.
2. Вторая операция — найти сумму первого числа в массиве и удалить его. Эта сумма равна 2, а массив после удаления будет равен $[2]$.
3. Третья операция — добавить пять чисел 4 в массив. После этой операции массив будет равен $[2, 4, 4, 4, 4, 4]$.
4. Четвертая операция — найти сумму первых трех чисел в массиве и удалить их. Эта сумма равна $2 + 4 + 4 = 10$, а массив после удаления будет равен $[4, 4, 4]$.

Задача F. Сортировка вагонов

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.

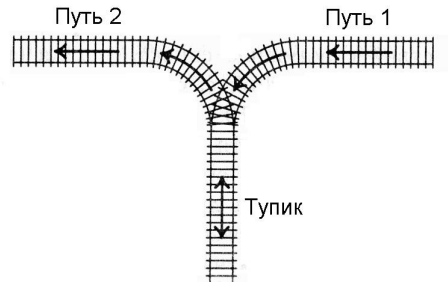

Рис. 1: схема путей

Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика).

Формат входных данных

Вводится число N — количество вагонов в поезде ($1 \leqslant N \leqslant 2000$). Дальше идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N, каждое из которых встречается ровно один раз.

Формат выходных данных

Если сделать так, чтобы вагоны шли в порядке от 1 до N, считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите действия, которые нужно проделать с поездом. В первой строке выведите количество действий, а затем сами действия. Каждое из них описывается двумя числами: типом и количеством вагонов:

- если нужно завезти с пути 1 в тупик K вагонов, должно быть выведено сначала число 1, а затем — число K ($K \geqslant 1$),
- если нужно вывезти из тупика на путь 2 K вагонов, должно быть выведено сначала число 2, а затем — число K ($K \geqslant 1$).

Если возможно несколько последовательностей действий, приводящих к нужному результату, выведите любую из них.

Если выстроить вагоны по порядку невозможно, выведите одно число 0.

Примеры

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

Задача G. Гоблины и очереди

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.6 секунд
Ограничение по памяти:	256 мегабайт

Гоблины Мглистых гор очень любях ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Формат входных данных

В первой строке входных данный записано число N ($1 \leqslant N \leqslant 10^5$) — количество запросов. Следующие N строк содержат описание запросов в формате:

- $+\ i$ — гоблин с номером i ($1 \leqslant i \leqslant N$) встаёт в конец очереди.
- $*\ i$ — привилегированный гоблин с номером i встает в середину очереди.
- `-` — первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Формат выходных данных

Для каждого запроса типа - программа должна вывести номер гоблина, который должен зайти к шаманам.

Примеры

стандартный ввод	стандартный вывод
7 + 1 + 2 - + 3 + 4 - -	1 2 3
2 * 1 + 2	

Задача H. Марсианская парикмахерская

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Пока я не поспал, «сегодня» не наступило

мистер Грин

В парикмахерской работает один мастер. Он тратит на одного клиента ровно 20 минут, а затем сразу переходит к следующему, если в очереди кто-то есть, либо ожидает, когда придет следующий клиент.

Даны времена прихода клиентов в парикмахерскую (в том порядке, в котором они приходили).

Также у каждого клиента есть характеристика, называемая *степенью нетерпения*. Она показывает, сколько человек может максимально находиться в очереди перед клиентом, чтобы он дождался своей очереди и не ушел раньше. Если в момент прихода клиента в очереди находится больше людей, чем степень его нетерпения, то он решает не ждать своей очереди и уходит. Клиент, который обслуживается в данный момент, также считается находящимся в очереди.

Требуется для каждого клиента указать время его выхода из парикмахерской.

Формат входных данных

В первой строке вводится натуральное число N, не превышающее 10^5 — количество клиентов.

В следующих N строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 16 000 000, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 10^5) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

Если для каких-то клиентов время окончания обслуживания одного клиента и время прихода другого совпадают, то можно считать, что в начале заканчивается обслуживание первого клиента, а потом приходит второй клиент.

Формат выходных данных

Выведите N пар чисел: времена выхода из парикмахерской 1-го, 2-го, ..., N-го клиента (часы и минуты). Если на момент прихода клиента человек в очереди больше, чем степень его нетерпения, то нужно считать, что время его ухода равно времени прихода.

Примеры

стандартный ввод	стандартный вывод
3 10 0 0 10 1 1 10 2 1	10 20 10 40 10 2
5 1 0 100 2 0 0 2 1 0 2 2 3 2 3 0	1 20 2 20 2 1 2 40 2 3

Задача I. Перлы и конвертер

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

Перлы — это мирная и первобытная раса, которая по вине человечества почти вымерла, а её оставшиеся представители дрейфовали по космосу. Прибыв на Альфу перлы познакомились с Валерианом и Лорелин и смогли наконец-то обзавестись конвертером жемчужин.

Конвертер — миленький зверек, который производит жемчужины k различных цветов. Для запуска двигателя космического корабля перлам нужен набор из k различных по цвету жемчужин. Конвертер производит одну жемчужину в секунду. Для эффективной работы двигателя нужно, чтобы в каждом наборе для любой пары жемчужин выполнялось условие, что разница во времени между появлением этих жемчужин не превосходит m секунд. Каждая жемчужина может входить только в один набор.

Конвертер произвел n жемчужин и устал. Помогите перлам узнать, наибольшее возможное число наборов жемчужин, которые они смогут собрать из имеющихся жемчужин.

Формат входных данных

В первой строке содержатся три целых числа n, m, k — количество жемчужин, произведенных конвертером, максимальный промежуток времени между появлением каждой пары жемчужин в одном наборе и количество различных цветов жемчужин соответственно ($1 \leqslant m \leqslant n \leqslant 10^5$, $1 \leqslant k \leqslant 10^5$).

В следующей строке содержатся n целых чисел a_i — цвет i-й появившейся жемчужины ($1 \leqslant a_i \leqslant k$).

Формат выходных данных

В первой строке выведите одно число x — наибольшее возможное число наборов жемчужин, которые перлы смогут собрать из имеющихся жемчужин.

В следующих x строках выведите по k целых чисел d_{ij} — номера жемчужин, входящих в i-й набор ($1 \leqslant d_{ij} \leqslant n$).

Если подходящих ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
6 2 3 1 2 2 1 3 3	1 4 3 5
2 1 2 1 1	0
5 2 3 1 2 2 2 3	0

Задача J. 2D динамический массив

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

Требуется реализовать структуру данных, которая будет поддерживать состояния n динамических массивов.

Вам дано q запросов трех типов:

- «1 i x» — добавить число $0 \leqslant x \leqslant 10^{18}$ в конец i-го массива. Гарантируется, что $1 \leqslant i \leqslant n$.
- «2 i» — вывести на экран и удалить последнее число из i-го массива. Гарантируется, что $1 \leqslant i \leqslant n$ и i-й массив в данный момент времени не пуст.
- «3 i j» — вывести на экран j-е число i-го массива. Гарантируется, что $1 \leqslant i \leqslant n, 1 \leqslant i \leqslant len$, где len — это размер массива в данный момент времени.

Формат входных данных

В первой строке вводятся два числа $1 \leqslant n, q \leqslant 10^6$ — количество массивов и запросов соответственно.

В последующие q строках вводятся запросы в выше описанном формате.

Формат выходных данных

Для каждого запроса второго и третьего типа выведите по одному числу.

Пример

стандартный ввод	стандартный вывод
2 10 1 1 2 1 2 2 1 1 3 1 2 0 1 2 1 2 1 3 1 1 1 1 4 2 2 2 2	3 2 1 0