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. Гистограмма

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Гистограмма является многоугольником, сформированным из последовательности прямоугольников, выровненных на общей базовой линии. Прямоугольники имеют равную ширину, но могут иметь различные высоты. Обычно гистограммы используются для представления дискретных распределений, например, частоты символов в текстах. Отметьте, что порядок прямоугольников очень важен. Вычислите область самого большого прямоугольника в гистограмме, который также находится на общей базовой линии.

Формат входных данных

В первой строке входного файла записано число N ($0 < N \leqslant 10^6$) — количество прямоугольников гистограммы. Затем следует N целых чисел $h_1 \ldots h_n$, где $0 \leqslant h_i \leqslant 10^9$. Эти числа обозначают высоты прямоугольников гистограммы слева направо. Ширина каждого прямоугольника равна 1.

Формат выходных данных

Выведите площадь самого большого прямоугольника в гистограмме. Помните, что этот прямоугольник должен быть на общей базовой линии.

Примеры

стандартный ввод	стандартный вывод
7 2 1 4 5 1 3 3	8
3 2 1 2	3
1 0	0

Задача D. Большой, белый, очень прямоугольный

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

В прямоугольной таблице клетки раскрашены в белый и черный цвета. Найти в ней прямоугольную область белого цвета, состоящую из наибольшего количества ячеек.

Формат входных данных

Во входных данных записана сначала высота N, а затем ширина M таблицы ($1 \leqslant N \leqslant 5000$, $1 \leqslant M \leqslant 5000$), а затем записано N строк по M чисел в каждой строке, где 0 означает, что соответствующая клетка таблицы выкрашена в белый цвет, а 1 — что в черный.

Формат выходных данных

В выходной файл вывести одно число — количество клеток, содержащихся в наибольшем по площади белом прямоугольнике.

Примеры

стандартный ввод	стандартный вывод
5 6 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	9
4 4 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0	4

Задача E. 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.

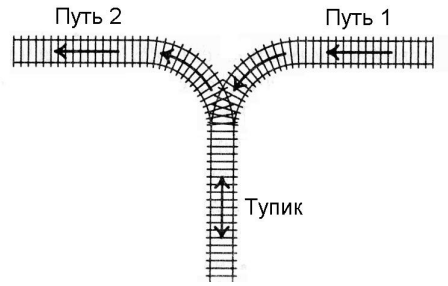

Рис. 1: схема путей

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 F. 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 G. Простая очередь

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

Реализуйте структуру данных «очередь». Напишите программу, содержащую описание очереди и моделирующую работу очереди, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:

- **push n** Добавить в очередь число n (значение n задается после команды). Программа должна вывести «ok».
- **pop** Удалить из очереди первый элемент. Программа должна вывести его значение.
- **front** Программа должна вывести значение первого элемента, не удаляя его из очереди.
- **size** Программа должна вывести количество элементов в очереди.
- **clear** Программа должна очистить очередь и вывести «ok».
- **exit** Программа должна вывести «bye» и завершить работу.

Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в очереди в любой момент не превосходит 100, все команды **pop** и **front** корректны, то есть при их исполнении в очереди содержится хотя бы один элемент.

Использовать стандартную очередь и дек из stl нельзя.

Формат входных данных

Вводятся команды управления очередью, по одной на строке.

Формат выходных данных

Требуется вывести протокол работы с очередью, по одному сообщению на строке.

Пример

стандартный ввод	стандартный вывод
size push 1 size push 2 size push 3 size exit	0 ok 1 ok 2 ok 3 bye