

Задача А. 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 В. Минимум на отрезке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность целых чисел длины N . По ней с шагом 1 движется «окно» длины K , то есть сначала в «окне» видно первые K чисел, на следующем шаге в «окне» уже будут находиться K чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

Формат входных данных

В первой строке входных данных содержатся два числа N и K ($1 \leq N \leq 150000$, $1 \leq K \leq 100000$, $K \leq N$) – длины последовательности и «окна», соответственно. На следующей строке находятся N чисел – сама последовательность. Числа последовательности не превосходят по модулю 10^5 .

Формат выходных данных

Выходные данные должны содержать $N - K + 1$ строк – минимумы для каждого положения «окна».

Пример

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

Задача С. 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.

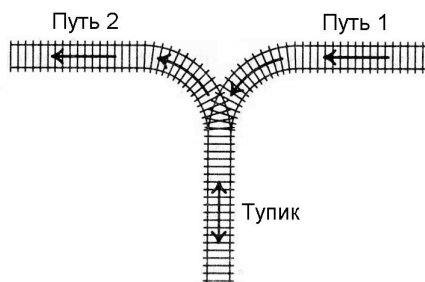


Рис. 1: схема путей

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 ($1 \leq N \leq 2000$). Далее 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 ($K \geq 1$),
- если нужно вывезти из тупика на путь 2 K вагонов, должно быть выведено сначала число 2, а затем — число K ($K \geq 1$).

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 D. 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 любят 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 данных записано число N ($1 \leq N \leq 10^5$) — количество запросов. Следующие N строк содержат описание запросов в формате:

- $+ i$ — гоблин с номером i ($1 \leq i \leq N$) 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 Е. Хорошие дни

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Вам дан массив целых неотрицательных чисел. Вы хотите найти в нем такой подотрезок (последовательность подряд идущих элементов), что в нем максимально возможная сумма, помноженная на минимум.

Поиск такого подотрезка является достаточно сложной задачей, поэтому мы приведем разбивку алгоритма по шагам. Не волнуйтесь, вам найдется, над чем подумать.

Итак, для начала требуется решить вспомогательную задачу - для каждого элемента массива найти ближайший слева элемент, меньший данного. То есть, посчитать $prev_i = j$, такой что $j < i$, $a_j < a_i$, $a_{j+1} \geq a_i$, $a_{j+2} \geq a_i, \dots a_i \geq a_i$.

Дальше, воспользовавшись таким алгоритмом, можно для каждого элемента i найти такой отрезок самой большой длины, что i является минимумом на всем этом отрезке. Ответом на задачу будет один из этих отрезков.

Для поиска суммы на отрезке можно воспользоваться, например, префиксными суммами.

Сделать поиск ближайшего элемента слева, меньше данного, можно сделать за один линейный проход со стеком — нужно только догадаться, что хранить в стеке, когда делать добавления, а когда удаления.

Формат входных данных

Первая строка входного файла содержит число n — количество элементов массива ($1 \leq n \leq 100\,000$).

Оставшаяся часть файла содержит n целых чисел a_1, a_2, \dots, a_n , все в пределах от 0 до 10^6 —. Числа во входном файле разделяются пробелами и переводами строки.

Формат выходных данных

В первой строке выходного файла выведите максимальную величину "сумма подотрезка, помноженная на минимум в этом подотрезке"

Примеры

стандартный ввод	стандартный вывод
6 3 1 6 4 5 2	60
4 1 2 1 2	6
3 2 1 2	5
3 2 2 1	8