

Задача А. Скобки

Имя входного файла: `brackets.in`
Имя выходного файла: `brackets.out`
Ограничение по времени: 0.5 second
Ограничение по памяти: 64 megabytes

Требуется определить, является ли правильной данная последовательность круглых, квадратных и фигурных скобок.

Формат входных данных

В единственной строке входного файла записано подряд N скобок ($1 \leq N \leq 10^5$).

Формат выходных данных

В выходной файл вывести «YES», если данная последовательность является правильной, и «NO» в противном случае.

Примеры

<code>brackets.in</code>	<code>brackets.out</code>
<code>()</code>	YES
<code>([])</code>	YES

Замечание

Скобочная последовательность называется правильной, если ее можно получить из какого-либо математического выражения вычеркиванием всех символов, кроме скобок.

Формальное определение правильной скобочной последовательности таково:

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

Задача В. Минимум на стеке

Имя входного файла: `stack.in`
Имя выходного файла: `stack.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам требуется реализовать структуру данных, выполняющую следующие операции:

1. Добавить элемент x в конец структуры.
2. Удалить последний элемент из структуры.
3. Выдать минимальный элемент в структуре.

Формат входных данных

В первой строке входного файла задано одно целое число n — количество операций ($1 \leq n \leq 10^6$). В следующих n строках заданы сами операции. В i -ой строке число t_i — тип операции (1, если операция добавления. 2, если операция удаления. 3, если операция минимума). Если задана операция добавления, то через пробел записано целое число x — элемент, который следует добавить в структуру ($-10^9 \leq x \leq 10^9$). Гарантируется, что перед каждой операцией удаления или нахождения минимума структура не пуста.

Формат выходных данных

Для каждой операции нахождения минимума выведите одно число — минимальный элемент в структуре. Ответы разделяйте переводом строки.

Примеры

stack.in	stack.out
8	-3
1 2	2
1 3	2
1 -3	
3	
2	
3	
2	
3	

Задача С. Минимум в окне

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

Дан ряд из N чисел. Требуется вывести минимумы из каждых K последовательных чисел.

Формат входных данных

В первой строке вводится одно натуральное число N , не превосходящее 100000. Во второй строке вводится одно натуральное число K , не превосходящее N . В следующих N строках вводится по одному натуральному числу, не превосходящему 100000.

Формат выходных данных

Выведите $N - K + 1$ чисел: минимум из первых K чисел, минимум из следующих K чисел (начиная со второго) и т.д.

Примеры

window.in	window.out
5	3
3	2
5	1
3	
5	
2	
1	

Задача D. Наибольший общий делитель

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Наибольшим общим делителем непустого набора натуральных чисел A называется максимальное натуральное число d , такое что оно является одновременно делителем всех чисел множества A .

Задан массив натуральных чисел $[a_1, a_2, \dots, a_n]$ и число k . Требуется выбрать в нем подмассив из k подряд идущих элементов $[a_l, a_{l+1}, \dots, a_{l+k-1}]$, чтобы их наибольший общий делитель был как можно больше, и вывести этот наибольший общий делитель.

Формат входных данных

Первая строка ввода содержит два целых числа n и k ($2 \leq n \leq 500\,000$, $2 \leq k \leq n$).

Вторая строка содержит n натуральных чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^{18}$).

Формат выходных данных

Выведите одно натуральное число — максимальное возможное значение наибольшего общего делителя элементов подмассива длины k заданного массива.

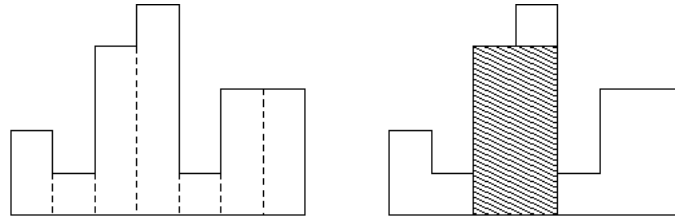
Примеры

стандартный ввод	стандартный вывод
10 4 2 3 4 8 12 6 12 18 4 3	6
2 2 12 18	6

Задача Е. Гистограмма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
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. Например, фигура на рисунке показывает гистограмму, которая состоит из прямоугольников с высотами 2, 1, 4, 5, 1, 3, 3. Все прямоугольники на этом рисунке имеют ширину, равную 1.



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. На рисунке справа заштрихованная фигура является самым большим выровненным прямоугольником на изображенной гистограмме.

Формат входных данных

В первой строке входного файла записано число N ($0 \leq N \leq 10^6$) — количество прямоугольников гистограммы. Затем следует N целых чисел $h_1, \dots, h_n$, где $0 \leq h_i \leq 10^9$. 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
4 1000 1000 1000 1000	4000

Задача F. Большой, белый, очень прямоугольный

Имя входного файла: `white.in`
Имя выходного файла: `white.out`
Ограничение по времени: 2 секунды
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 входном файле записана сначала высота N , а затем ширина M таблицы ($1 \leq N \leq 5000$), ($1 \leq M \leq 5000$), 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

white.in	white.out
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 G. Программирование квадрокоптеров

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

Школьники готовятся к участию в соревновании по программированию квадрокоптеров. Квадрокоптер, который используется в соревновании, может выполнять две команды: подняться вверх на 1 метр и опуститься вниз на 1 метр. Команда подъёма обозначается символом «(», а команда спуска — символом «)».

Программа для квадрокоптера представляет собой последовательность команд. Программа считается корректной, если, начав её исполнение на уровне земли и выполнив последовательно все команды, квадрокоптер снова оказывается на уровне земли. При этом в процессе выполнения программы квадрокоптер не должен пытаться опуститься ниже уровня земли.

Например, следующие программы являются корректными: «()()», «((()))». Программа «(((» не является корректной, поскольку квадрокоптер завершает её выполнение на высоте 3 метра над уровнем земли, программа «()»(» также не является корректной, поскольку при выполнении третьей команды квадрокоптер пытается опуститься ниже уровня земли.

Участник соревнования написал корректную программу для квадрокоптера, состоящую из n команд, пронумерованных от 1 до n . Он загрузил её в память квадрокоптера для демонстрации во время соревнования. К сожалению, после загрузки программы в память квадрокоптера участник случайно удалил её на своём компьютере, а квадрокоптер не позволяет выгрузить программу из своей памяти.

К счастью, квадрокоптер поддерживает специальный режим отладки программы. В этом режиме квадрокоптер с загруженной в него программой может отвечать на специальные запросы. Каждый запрос представляет собой два целых числа: l и r , $1 \leq l \leq r \leq n$. В ответ на запрос квадрокоптер сообщает, является ли фрагмент загруженной в него программы, состоящий из команд с l -й по r -ю включительно, корректной программой для квадрокоптера, либо нет. Участник хочет с помощью режима отладки восстановить загруженную в квадрокоптер программу.

Требуется написать программу-решение, которая взаимодействует с программой жюри, моделирующей режим отладки квадрокоптера, и в итоге восстанавливает загруженную в квадрокоптер программу.

Формат входных данных

Это интерактивная задача.

Сначала на вход подаётся целое число n — количество команд в программе квадрокоптера ($2 \leq n \leq 50000$).

Для каждого теста жюри зафиксировано число k — максимальное количество запросов. Гарантируется, что k запросов достаточно, чтобы решить задачу. Это число не сообщается программе-решению. Ограничения k в различных подзадачах приведены в таблице системы оценивания. Если программа-решение делает более k запросов к программе жюри, то на этом тесте она получает в качестве результата тестирования «Неверный ответ».

Чтобы сделать запрос, программа-решение должна вывести строку вида «? l r », где l и r — целые положительные числа, задающие фрагмент программы квадрокоптера ($1 \leq l \leq r \leq n$).

В ответ на запрос программы-решения программа жюри подаёт ей на вход либо строку «Yes», либо строку «No», в зависимости от того, является ли запрошенный фрагмент программы квадрокоптера корректной программой.

Если программа-решение определила ответ на задачу, то она должна вывести строку «! c_1 c_2 ... c_n », где символ c_i задаёт i -ю команду в программе квадрокоптера и равен либо «(», либо «)».

После этого программа-решение должна завершиться.

Гарантируется, что в каждом тесте программа в памяти квадрокоптера является фиксированной корректной программой, которая не меняется в зависимости от запросов, произведённых программой-решением.

Формат выходных данных

Примеры

стандартный ввод	стандартный вывод
4 <ожидание ответа> Yes <ожидание ответа> No <ожидание ответа> Yes <ожидание ответа> Yes <ожидание ответа>	<ожидание ввода> ? 1 4 <ожидание ввода> ? 1 3 <ожидание ввода> ? 1 2 <ожидание ввода> ? 3 4 <ожидание ввода> ! () ()
6 <ожидание ответа> Yes <ожидание ответа> No <ожидание ответа> Yes <ожидание ответа>	<ожидание ввода> ? 3 4 <ожидание ввода> ? 1 2 <ожидание ввода> ? 2 5 <ожидание ввода> ! ((()))

Задача Н. Минимумы в подматрицах

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

Дана матрица $n \times n$, состоящая из целых чисел. Для каждой её подматрицы размера $L \times L$ найдите минимум в этой подматрице. Подматрицей здесь называется “подпрямоугольник”.

Внимание. Решение должно работать за $O(n^2)$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и L ($1 \leq L \leq n \leq 1000$). Далее в n строках идет описание матрицы $n \times n$, по n чисел в каждой строке. Все числа в матрицы целые, от -10^9 до 10^9 .

Формат выходных данных

Выведите $n - L + 1$ строку по $n - L + 1$ числу в каждой. j -е число в i -й строке должно быть равно минимуму в подматрице размера $L \times L$ с левым верхним углом на пересечении i -й строки и j -го столбца исходной матрицы.

Примеры

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

Задача I. Интеллектуальный отпуск

Имя входного файла: e.in
Имя выходного файла: e.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Туристическая отрасль в этом сезоне столкнулась с серьёзными сложностями. Добросовестные туроператоры ищут новые рекламные ходы для продажи своих туров. Как известно, наиболее благоприятная для отдыха погода меняется плавно, причём не только от одного дня к другому, но и в течение суток.

Для большинства туристических направлений есть многолетние посекундные результаты измерений различных климатических параметров, например, температуры или влажности. У каждого человека своё понимание того, насколько различными могут быть подобные значения во время отпуска, но всех интересуют непрерывные туры как можно большей продолжительности.

Пусть мы зафиксировали туристическое направление и некоторый климатический параметр. Будем называть изменчивостью тура разницу между максимальным и минимальным значением выбранного параметра за всё время поездки. Для каждого туриста известно максимальное приемлемое значение изменчивости k_i .

Даны результаты измерений некоторого климатического параметра на одном из курортов и значения k_i для нескольких туристов. Требуется для каждого из них определить максимальный диапазон, подходящий для отпуска.

Формат входных данных

В первой строке входного файла находится целое число N ($1 \leq N \leq 600\,000$) — количество сделанных измерений. Во второй строке — N целых чисел, по модулю не превосходящих 10^9 — данные посекундных измерений.

В третьей строке входного файла находится число M ($1 \leq M \leq 100$) — количество туристов, для которых необходимо найти оптимальный диапазон. В четвёртой строке — M целых чисел $k_1, k_2, \dots, k_M$ ($0 \leq k_i \leq 10^9$) — максимальная возможная разница между выбранным климатическим параметром в непрерывном диапазоне дней для каждого из туристов.

Формат выходных данных

В выходной файл для каждого из M запросов в отдельной строке выведите два числа: номер первого измерения диапазона и номер последнего измерения, входящего в диапазон. Нумерация измерений ведётся с единицы. Если для некоторого туриста существует несколько подходящих диапазонов максимальной длины, выведите границы любого из них.

Примеры

e.in	e.out
7	3 5
10 1 10 12 11 1 11	4 5
2	
2 1	
9	3 4
1 5 2 3 6 4 7 8 9	1 9
6	7 9
1 10 2 4 5 0	2 6
	1 6
	1 1