

Задача А. Запросы сумм

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

Задана последовательность из n целых чисел. Также заданы k запросов вида $i\ j$. Для каждого запроса необходимо вывести сумму элементов последовательности, начиная с a_i до a_j включительно.

Формат входных данных

В первой строке задано натуральное число n — количество элементов в последовательности ($2 \leq n \leq 10^5$). Во второй строке заданы n целых чисел через пробел — элементы последовательности ($|a_i| \leq 10^9$). В третьей строке задано число k ($1 \leq k < 10000$). В следующих k строках записано по два натуральных числа i и j ($1 \leq i < j \leq n$).

Формат выходных данных

Для каждого запроса выведите сумму элементов последовательности, начиная с a_i до a_j включительно.

Пример

стандартный ввод	стандартный вывод
5	3
1 0 -7 2 7	1
3	2
1 5	
1 2	
3 5	

Задача В. Максимальная разность

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

Задана последовательность из n целых чисел. Необходимо найти два таких элемента, что разность между правым и левым значениями будет максимальна.

Формат входных данных

В первой строке задано натуральное число n — количество элементов в последовательности ($2 \leq n \leq 10^5$). В следующей строке заданы n целых чисел через пробел — элементы последовательности ($|a_i| \leq 10^9$)

Формат выходных данных

В первой строке выведите два числа i и j — номера элементов a_i и a_j в последовательности таки, что $a_j - a_i$ максимально. Во второй строке выведите самую максимальную разность $a_j - a_i$.

Пример

стандартный ввод	стандартный вывод
5 1 3 8 -2 3	1 3 7

Задача С. Максимальная разность - 2

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

Задана последовательность из n целых чисел. Необходимо найти два таких элемента, что разность между правым и левым значениями будет максимальна, при этом между ними находится не менее, чем k элементов.

Формат входных данных

В первой строке задано натуральное число n — количество элементов в последовательности ($3 \leq n \leq 10^5$). Во второй строке задано число k ($1 \leq k \leq 10$). Гарантируется, что $n > k$.

В следующей строке заданы n целых чисел через пробел — элементы последовательности ($|a_i| \leq 10^9$).

Формат выходных данных

В первой строке выведите два числа i и j — номера элементов a_i и a_j в последовательности такие, что $a_j - a_i$ максимально, при этом между a_i и a_j должно быть не менее чем k элементов.

Во второй строке выведите саму максимальную разность $a_j - a_i$.

Если есть несколько одинаковых пар, то выведите любую.

Примеры

стандартный ввод	стандартный вывод
6 2 1 3 8 -2 3 15	1 6 14
7 2 -1 -2 -3 -4 -5 -6 -7	1 4 -3

Задача D. 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 Е. 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 F. Максимальная сумма

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

Задана последовательность из n целых чисел. В этой последовательности найдите непрерывный подотрезок, сумма чисел в котором максимальна. Примечание. Фактически требуется найти такие i и j ($i \leq j$), что сумма всех элементов массива от a_i до a_j включительно будет максимальна.

Формат входных данных

В первой строке задано натуральное число n — количество элементов в последовательности ($2 \leq n \leq 10^5$). В следующей строке заданы n целых чисел через пробел — элементы последовательности ($|a_i| \leq 10^9$)

Формат выходных данных

Выдайте пару искомых значений индексов. Если таких пар несколько, то j должно быть минимально возможным, а при равных j значение i должно быть максимально возможным.

Пример

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

Задача G. Гемоглобин

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Каждый день к Грегори Хаусу приходит много больных, и у каждого измеряется уровень гемоглобина в крови. Данные по всем пациентам заносятся в базу данных.

Но волчанка попадается один раз на миллион, а работать с остальными неинтересно. Чтобы Хаус не выгонял больных, Кадди иногда запрашивает статистику по k последним больным: ей хочется знать сумму их уровня гемоглобина.

Также Хаус — мизантроп: он смотрит уровень гемоглобина больного, который поступил к нему позже всех, и, видя, что это точно не волчанка, выписывает его из больницы и удаляет информацию о нем из базы.

Автоматизацию процесса Хаус поручил Чейзу. Но Чейз почему-то не справился с этой задачей и попросил вас ему помочь.

Формат входных данных

Первой строкой входного файла задано число n ($1 \leq n \leq 100\,000$) — число обращений к базе данных. Запросы к базе выглядят следующим образом: «+ x » ($1 \leq x \leq 10^9$) — добавить пациента с уровнем гемоглобина x в базу, «-» — удалить последнего пациента из базы, «? k » ($1 \leq k \leq 100\,000$) — вывести суммарный гемоглобин последних k пациентов. Гарантируется, что k не превосходит число элементов в базе. Также гарантируется, что запросов на удаление к пустой базе не поступает. Перед началом работы база данных пуста.

Формат выходных данных

Для каждого запроса «-» вывести уровень гемоглобина в крови пациента, а для каждого запроса «? k » — суммарный гемоглобин у последних k поступивших пациентов. Ответы выводите в порядке поступления запросов.

Примеры

стандартный ввод	стандартный вывод
7	5
+1	3
+2	2
+3	1
?2	
-	
-	
?1	

Задача Н. 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 I. Стена Тысячи Башен

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

В королевстве Робто существует Великая Стена, состоящая из n башен, выстроенных в ряд. Каждая башня обладает своей прочностью.

С севера надвигается армия вражеского королевства. Чтобы защитить свои земли, вы, как главный стратег в Робто, предлагаете выбрать один непрерывный участок стены и сосредоточить на нём силы. Однако эффективность выбранного участка определяется двумя факторами: длиной участка (сколько башен в него входит) и прочностью самой слабой башни на этом участке. Слишком длинный участок может оказаться уязвим из-за слабого звена, а слишком короткий — не позволит разместить достаточно войск.

Вам необходимо выбрать такой непрерывный участок стены, чтобы его эффективность (то есть минимум из длины участка и прочностью самой слабой башни на участке) была максимальной.

Формат входных данных

Первая строка содержит одно целое число n — количество башен ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$), где a_i — прочность i -й башни.

Формат выходных данных

Выведите одно число — максимальную возможную эффективность выбранного участка.

Примеры

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

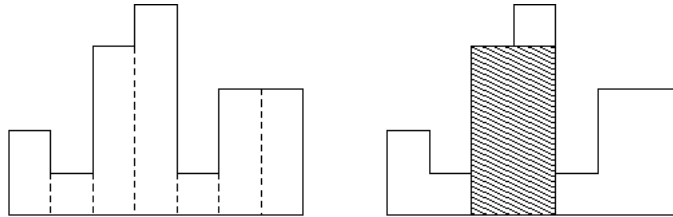
Замечание

В эту задачу нужно сдать решение за $\mathcal{O}(n)$. Решения, не работающие за линейное время, получают **RJ**.

Задача J. 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 К. Город Че

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

В центре города Че есть пешеходная улица — одно из самых популярных мест для прогулок жителей города. По этой улице очень приятно гулять, ведь вдоль улицы расположено n забавных памятников.

Девочке Маше из города Че нравятся два мальчика из ее школы, и она никак не может сделать выбор между ними. Чтобы принять окончательное решение, она решила назначить обоим мальчикам свидание в одно и то же время. Маша хочет выбрать два памятника на пешеходной улице, около которых мальчики будут ее ждать. При этом она хочет выбрать такие памятники, чтобы мальчики не увидели друг друга. Маша знает, что из-за тумана мальчики увидят друг друга только в том случае, если они будут на расстоянии не более r метров.

Маше заинтересовалась, а сколько способов есть выбрать два различных памятника для организации свиданий.

Формат входных данных

В первой строке входного файла находятся два целых числа n и r ($2 \leq n \leq 300\,000$, $1 \leq r \leq 10^9$) — количество памятников и максимальное расстояние, на котором мальчики могут увидеть друг друга.

Во второй строке задано n положительных чисел $d_1, \dots, d_n$, где d_i — расстояние от i -го памятника до начала улицы. Все памятники находятся на разном расстоянии от начала улицы. Памятники приведены в порядке возрастания расстояния от начала улицы ($1 \leq d_1 < d_2 < \dots < d_n \leq 10^9$).

Формат выходных данных

Выведите одно число — число способов выбрать два памятника для организации свиданий.

Примеры

<code>che.in</code>	<code>che.out</code>
4 4 1 3 5 8	2

Замечание

В приведенном примере Маша может выбрать памятники 1 и 4 или памятники 2 и 4.

Задача L. Разрушение моста

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

Давным-давно между Берляндией и Флатландией был построен мост состоящий из n секций, каждая из которых имеет собственный показатель прочности. К сожалению, совсем недавно m -я по счёту секция моста разрушилась и мост естественным образом разбился на две части. Первые $m - 1$ секций всё ещё были соединены с Берляндией, а оставшиеся $n - m$ секций – с Флатландией.

Лучшие инженеры двух стран решили, что необходимо реконструировать мост, а для этого старый необходимо частично разрушить. После долгих обсуждений инженеры придумали следующий метод частичного разрушения моста. Каждая из стран в определённом порядке будет производить несколько «подрывов». «Подрывом» называется разрушение самой слабой секции моста, достижимой из определённой страны. То есть, если «подрыв» производит Берляндия, то она должна уничтожить самую слабую секцию моста среди первых $k - 1$, где k — минимальный номер уже разрушенной секции моста. Если существует несколько самых слабых секций, то выбирается наиболее далёкая от материка взрывающей страны.

Вам известен порядок, в котором страны будет производить подрыв. Помогите инженерам, назвав номера уничтожаемых обеими странами секций в хронологическом порядке. Гарантируется, что перед каждым «подрывом» существует хотя бы одна секция, граничащая со страной, производящей «подрыв».

Формат входных данных

Первая строка содержит целые числа n и m ($2 \leq n \leq 3 \cdot 10^5, 1 \leq m \leq n$) — количество секций моста до разрушения и первая разрушенная секция моста.

Следующая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10^9$) — прочность i секции моста соответственно.

Третья строка s описывает порядок «подрывов» и содержит только символы «В» и «F», которые обозначают Берляндию и Флатланию соответственно. s_i символ обозначает, кто именно будет производить i -й в хронологическом порядке подрыв.

Формат выходных данных

Выведите в одну строку n чисел через пробел. i -м числом в строке должен быть номер секции, которая будет взорвана i -й по очереди.

Система оценки

Помимо тестов из условия, данная задача содержит 50 тестов, каждый из которых оценивается независимо. За каждый успешно пройденный тест вы получите 2 балла.

Пример

стандартный ввод	стандартный вывод
10 4 3 1 9 2 10 3 1 4 2 8 BFFB	2 7 9 1

Задача М. 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.

В тестах $k = 100000$, то есть, разрешается произвести не более 100000 запросов.

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 N. Атака титанов

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

В этой задаче мы хотим увидеть решение *ЛИНЕЙНОЙ* сложности (т.е. мы не хотим что-то похожее на $O(n \log n)$)

Как известно из манги и аниме «Атака Титанов», выжившие за стенами люди ведут войну с огромными существами - Титанами.

В 843 году отряд разведки обнаружил отряд титанов, движущийся к внешней стене. Для оперативной ликвидации угрозы к защитникам стены (гарнизону) послали гонца с описанием роста каждого из титанов. Благодаря brave инженерам людей были созданы пушки, уничтожающие титанов с одного выстрела, но, к сожалению, наука далека от совершенства, поэтому у пушек есть 2 проблемы:

1. Одна пушка может уничтожать только титанов определённого роста, который заранее неизвестен
2. Идеально откалибровать пушку невозможно, поэтому она устанавливается на определённую высоту и может подниматься не больше чем на **коэффициент калибровки**.

Для примера, если **коэффициент калибровки** равен 5, и пушка установлена на высоту 15, она сможет уничтожать только титанов какой-то конкретной высоты от 15 до 20 включительно (например, 17).

У гарнизона существуют пушки для каждой возможной высоты, но, к сожалению, из-за срочности они могут подвезти к стене только одну из них. Благодаря знанию структур данных защитники стены смогли выбрать пушку, которая сможет **гарантированно** уничтожить как можно больше титанов, и пережили атаку. А сможете ли Вы?

Кроме того, чем выше пушка, тем больше работы предстоит гарнизону, поэтому сообщайте самую низкую пушку из тех, что дают лучший результат.

Формат входных данных

В первой строке подаётся два числа: N - количество титанов ($1 \leq N \leq 10^6$), k - коэффициент калибровки ($1 \leq k \leq 10^6$). Во второй строке находятся N чисел - рост всех каждого из титанов. Рост каждого титана - целое число в диапазоне от 1 до 10^6 включительно

Формат выходных данных

В первой строке выведите одно число: высоту на которую должна быть установлена пушка (минимальную среди лучших)

Во второй строке выведите одно число: количество титанов которое гарантированно будет уничтожено

Примеры

стандартный ввод	стандартный вывод
10 3 1 4 4 3 1 4 4 2 2 3	1 2
3 1 24 18 5	1 0