

Задача А. Сумма простая

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вам нужно научиться отвечать на запрос «сумма чисел на отрезке».

Массив не меняется. Запросов много. Отвечать на каждый запрос следует за $\mathcal{O}(1)$.

Формат входных данных

Размер массива — n и числа x, y, a_0 , порождающие массив a : $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$

Далее следует количество запросов m и числа z, t, b_0 , порождающие массив b : $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$.

Массив c строится следующим образом: $c_i = b_i \bmod n$.

Запросы: i -й из них — найти сумму на отрезке от $\min(c_{2i}, c_{2i+1})$ до $\max(c_{2i}, c_{2i+1})$ в массиве a .

Ограничения: $1 \leq n \leq 10^7$, $0 \leq m \leq 10^7$. Все числа целые от 0 до 2^{16} . t может быть равно -1 .

Формат выходных данных

Выведите сумму всех сумм.

Пример

стандартный ввод	стандартный вывод
3 1 2 3 3 1 -1 4	23

Замечание

$a = \{3, 5, 7\}$, $b = \{4, 3, 2, 1, 0, 2^{30} - 1\}$, $c = \{1, 0, 2, 1, 0, 0\}$,
запросы = $\{[0, 1], [1, 2], [0, 0]\}$, суммы = $\{8, 12, 3\}$.

Задача В. Максимальная сумма

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

В этой задаче вам требуется найти непустой отрезок массива с максимальной суммой.

Формат входных данных

В первой строке входных данных записано единственное число n ($1 \leq n \leq 3 \cdot 10^5$) — размер массива.

Во второй строке записано n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — сам массив.

Формат выходных данных

Выведите одно число — максимальную сумму на отрезке в данном массиве.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	10
4 5 4 -10 4	9

Задача С. Прибавляем, суммируем

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Есть массив целых чисел длины $n = 2^{24}$, изначально заполненных нулями. Вам нужно сперва обработать m случайных запросов вида «прибавление на отрезке». Затем обработать q случайных запросов вида «сумма на отрезке».

Формат входных данных

На первой строке числа m, q ($1 \leq m, q \leq 2^{24}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
0. unsigned int a, b; // даны во входных данных
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
```

Каждый запрос первого вида генерируется следующим образом:

```
1. add = nextRand(); // число, которое нужно прибавить
2. l = nextRand();
3. r = nextRand();
4. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Каждый запрос второго вида генерируется следующим образом:

```
1. l = nextRand();
2. r = nextRand();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются запросы первого вида, затем второго.

Формат выходных данных

Выведите сумму ответов на все запросы второго типа по модулю 2^{32} .

Примеры

стандартный ввод	стандартный вывод
5 5 13 239	811747796
10 10 239017 170239	3460675938

Задача D. Возрастающий массив

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

В свободное время Валериан любит решать и задавать Лорелин разные задачи. Сегодня он придумал такую задачу: дан массив из n целых чисел a_1, a_2, \dots, a_n . Можно некоторые элементы массива a_i заменить на $-a_i$. Требуется сделать массив неубывающим или сказать, что это невозможно.

Формат входных данных

В первой строке дано одно целое число n — количество элементов в массиве ($1 \leq n \leq 100\,000$).
В следующей строке дано n целых чисел a_i — исходный массив ($|a_i| \leq 100\,000$).

Формат выходных данных

Если нельзя заменить некоторые элементы a_i на $-a_i$, чтобы массив стал неубывающим, выведите «No».

Иначе, в первой строке выведите «Yes». Во второй строке выведите n чисел b_i , которые образуют неубывающий массив, и для всех $1 \leq i \leq n$ выполняется $b_i = a_i$ или $b_i = -a_i$.

Система оценки

Эта задача состоит из четырех подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех необходимых подзадач. Номера необходимых подзадач также указаны в таблице.

Обратите внимание, что некоторые **тесты из условия** не подходят под ограничения некоторых подзадач, однако они **обязательно должны быть пройдены** для того, чтобы решение было принято на проверку.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи
1	20	$n \leq 20$	—
2	25	$n \leq 1\,000, 0 \leq a_i$	—
3	25	$n \leq 1\,000, a_i \leq 0$	—
4	30	нет дополнительных ограничений	1, 2, 3

Примеры

стандартный ввод	стандартный вывод
5 1 -1 -2 3 6	Yes -1 -1 2 3 6
3 1 1 0	Yes -1 -1 0

Задача Е. Оптимальная пара

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

Дан массив a , состоящий из n целых чисел. Найдите такую пару (i, j) , что $1 \leq i < j \leq n$ и значение $a_j - a_i$ минимально.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 10^5$) — количество элементов в массиве.
Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы массива.

Формат выходных данных

Выведите через пробел два целых числа i и j , такие что $1 \leq i < j \leq n$ и $a_j - a_i$ минимально.
Если существует несколько оптимальных пар (i, j) , выведите любую из них.

Примеры

стандартный ввод	стандартный вывод
9 2 1 5 -2 0 -1 10 5 3	7 9
2 1 5	1 2
5 5 4 3 2 1	1 5

Замечание

В первом примере оптимальными являются пары $(7, 9)$ и $(3, 4)$.

Задача F. Лето в Сочи

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

Илья очень хочет отдохнуть во время летних каникул в Сочи у бабушки. Лето этого года состоит из N дней. Илья не любит жару, поэтому с интересом ознакомился с прогнозом температуры на каждый летний день: t_1, t_2, \dots, t_N . Родители отпускают его к бабушке с тем расчетом, что он проведет у нее как можно больше времени. Разумеется, Илья хочет выбрать такой период отдыха в Сочи, средняя температура за который будет наименьшей. Если же таких периодов может быть несколько, то предпочтение он отдаст самому большому по количеству дней периоду времени. Илья соскучился и хочет попасть к бабушке как можно раньше. Поэтому если таких периодов будет несколько, то он хотел бы выбрать тот, который начинается раньше других.

Составьте программу, которая определит день начала и окончания отдыха, а также среднюю температуру воздуха за этот период.

Формат входных данных

В первой строке дано число N — количество дней этого лета ($1 \leq N \leq 3 \cdot 10^5$). В следующей строке даны t_1, t_2, \dots, t_N — температура в каждый из дней ($1 \leq t_i \leq 10^9$).

Формат выходных данных

В единственной строке выведите два целых числа — день, в который Илья приедет, а также сколько дней он проведет у бабушки.

Пример

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

Задача G. M — многомерность

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

Многие в детстве играют с кубиками, затем все в школе изучают геометрию и встречаются с такими простыми объектами, как параллелепипеды. Но ведь изучать геометрию в трехмерном пространстве — это так скучно! Даже четырехмерным пространством уже никого не удивишь! Поэтому в этой задаче мы предлагаем вам изучить параллелепипеды в M -мерном пространстве. Чувствуете, как интересно?

Определим M -мерный параллелепипед как набор отрезков $[a_1, b_1], [a_2, b_2], \dots, [a_M, b_M]$, где $a_i < b_i$ для всех $i = 1 \dots M$. Для простоты будем считать, что все a_i и b_i являются целыми.

Скажем, что точка с координатами (x_1, x_2, \dots, x_M) лежит внутри параллелепипеда, если выполнены неравенства:

$$\begin{aligned} a_1 &\leq x_1 \leq b_1, \\ a_2 &\leq x_2 \leq b_2, \\ &\dots \\ a_M &\leq x_M \leq b_M. \end{aligned}$$

Вам даны N M -мерных параллелепипедов. Требуется посчитать, сколько точек с целочисленными координатами лежат внутри ровно $N - 1$ параллелепипеда. Так как ответ может быть большим, выведите остаток от деления количества точек на число 998 244 353.

Формат входных данных

В первой строке записаны два числа N и M ($2 \leq N \leq 2 \cdot 10^5, 1 \leq M \leq 2 \cdot 10^5, 2 \leq N \cdot M \leq 2 \cdot 10^5$) — количество параллелепипедов и размерность пространства соответственно.

Каждая из следующих N строк задает параллелепипед. В каждой строке через пробел записаны $2 \cdot M$ чисел в следующем порядке: $a_1, b_1, a_2, b_2, \dots, a_M, b_M$ ($-10^6 \leq a_i < b_i \leq 10^6$ для всех i).

Формат выходных данных

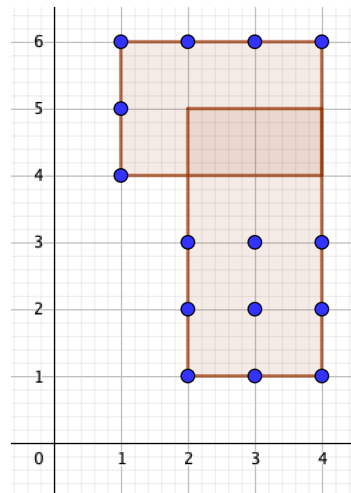
Выведите одно число — остаток от деления количества точек, лежащих внутри ровно $N - 1$ параллелепипеда, на число 998 244 353.

Примеры

стандартный ввод	стандартный вывод
2 2 2 4 1 5 1 4 4 6	15
4 1 1 6 2 4 6 7 2 9	4

Замечание

Рисунок к первому примеру, в котором даны два прямоугольника. Необходимо посчитать все целочисленные точки, которые лежат внутри (или на границе) ровно одного прямоугольника. В данном примере таких точек 15. Все эти точки отмечены на рисунке.



Во втором примере $M = 1$, значит параллелепипеды являются обыкновенными отрезками на прямой.

Задача Н. Антипа и бухгалтерия

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

Каждый месяц князь Андрей (если вы не знали, то князь Андрей — хороший друг Антипы) обходит захваченные территории, собирая с них дань.

В подчинении Андрея находится n городов, каждый из которых выплачивает князю дань. Города пронумерованы от 1 до n . Город с номером i должен заплатить дань, отдав все золотые монеты, которые в нем есть. Князь обходит города в порядке их нумерации от 1 до n , и дань, заплаченную каждым городом, складывает в большой мешок. Изначально мешок пуст.

Антипа решил помочь Андрею и нанялся к нему бухгалтером. После каждого посещения города Антипа записывает на отдельный листок общее количество монет в большом мешке князя. Таким образом, после обхода у него имеется n листочков, на i -м из которых написано число i и суммарное количество монет, собранное с городов от первого до i -го.

К сожалению, из-за сильного ветра некоторые листочки Антипы улетели. Сегодня князь решил разобраться, почему в казне так мало денег, и взял листочки Антипы, которые не были унесены ветром. У Князя закралось подозрение, что Антипа мог ошибиться в своих записях (подозреваю, что это из-за того, что его особо никто не учил считать и писать).

Андрею точно известно, что каждый город заплатил в качестве дани хотя бы одну золотую монету. Помогите ему понять по оставшимся записям, ошибся ли бухгалтер Антипа в каких-то из них, а если не ошибся, то подскажите, сколько монет мог заплатить каждый из городов.

Формат входных данных

В первой строке входных данных содержится число n ($1 \leq n \leq 10^5$) — количество городов в подчинении у Андрея.

Во второй строке содержатся n целых чисел, разделенных одним пробелом, i -е из которых равно -1 , если листочек с записью суммарного дохода после посещения i -го города был потерян, а иначе это число равно числу, записанному на i -м листке.

Все числа во входном файле от -1 до 10^9 .

Формат выходных данных

Если в записях Антипы точно была допущена ошибка, выведите «NO».

В противном случае, в первой строке выведите «YES», а во второй строке выведите n натуральных чисел от 1 до 10^9 , i -е из которых равняется количеству монет, полученных от города с номером i . Если есть несколько возможных ответов, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
5 1 3 -1 10 -1	YES 1 2 3 4 5
3 10 -1 4	NO

Задача I. Межпланетный лифт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Недавно у Мстителей появилась новая штаб-квартира. Первым делом Тони Старк решил установить там лифт, ведь не все умеют летать. Однако он не уследил за Халком, и тот в порыве гнева поставил два лифта вместо одного. Позже он объяснил это тем, что два лифта могут перевозить в два раза больше людей, но не учел, что лифты не могут проходить друг сквозь друга.

Штаб-квартиру мстителей можно разбить на этажи. Дело произошло не на Земле, поэтому тут очень много этажей (нет никаких ограничений в передвижении лифта), и даже есть отрицательные! У каждого лифта есть своя программа — строка длины M , состоящая из нулей и единиц. Если i -й символ строки равен нулю, то эта команда опускает лифт вниз на один этаж, иначе — поднимает вверх (аналогично на один этаж). Если лифты столкнутся, то они сломаются. На данный момент лифты не двигаются, а после запуска поедут одновременно и остановятся по истечении M секунд.

Железный человек отходил спасать мир, поэтому не заметил этой проблемы. На данный момент лифты находятся на этажах P_1, P_2 соответственно. До запуска лифтов Тони может успеть исправить суммарно не более K команд. Другими словами, Железный человек может не более K раз выбрать любую команду одного из лифтов и инвертировать ее — если команда была равна единице, Тони заменит ее на ноль, а если она была равна нулю, то он заменит ее на единицу. Тони может инвертировать команды как первого, так и второго лифта несколько раз.

Необходимо написать программу, которая определяет, достаточно ли K исправлений команд, чтобы лифты не столкнулись.

Формат входных данных

В первой строке вводятся числа M, K ($1 \leq M \leq 10^5, 0 \leq K \leq 2 \cdot 10^5$) — длина программ и количество секунд, оставшихся у Железного человека.

Во второй и третьей строках вводятся последовательности длины M , состоящие из нулей и единиц, — программы первого и второго лифтов.

В четвертой строке вводятся два числа P_1, P_2 ($-10^9 \leq P_1, P_2 \leq 10^9, P_1 \neq P_2$) — позиции первого и второго лифтов соответственно

Формат выходных данных

Если невозможно исправить не более K символов так, чтобы лифты не сломались, то в единственной строке выведите «NO» (без кавычек).

Иначе в первой строке выведите «YES» (без кавычек), во второй — исправленную первую программу, а в третьей — исправленную вторую.

Если при текущих программах лифты не сломаются, то можно вывести программы без изменений.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения		Необходимые подзадачи	Информация о проверке
		M	K		
0	0	Тесты из условия		—	полная
1	10	$1 \leq M \leq 8$	$0 \leq K \leq 2 \cdot 10^5$	—	полная
2	10	$1 \leq M \leq 10^5$	$K = 0$	—	первая ошибка
3	10	$1 \leq M \leq 10^5$	$K = M$	—	первая ошибка
4	30	$1 \leq M \leq 100$	$0 \leq K \leq 2 \cdot 10^5$	1	первая ошибка
5	40	$1 \leq M \leq 10^5$	$0 \leq K \leq 2 \cdot 10^5$	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 10 10101 01010 1 10	YES 00000 11111
5 0 00000 11111 2 1	NO

Задача J. Очередная задача про игру с камнями

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

В результате неудачного стечения обстоятельств Миша оказался на необитаемом острове. Первым делом он, разумеется, выложил из камней большую надпись S.O.S. на пляже. Однако, до прибытия помощи ему нужно чем-то развлечься, поэтому он решил поиграть с оставшимися камнями.

Миша выложил все оставшиеся у него камни в n кучек таким образом, что в i -й кучке оказалось ровно r_i камней. После этого мальчик решил взять из каждой кучки некоторое количество камней, чтобы были выполнены следующие условия:

- Из i -й кучки Миша должен взять не менее, чем l_i и не более, чем r_i камней;
- Суммарное количество взятых камней должно быть равно s .

После того, как Миша справился с данным заданием он задумался, сколькими способами он может взять камни из кучек таким образом, чтобы описанные условия были выполнены. А именно, для каждой кучки i он хочет вычислить, сколькими способами он может выбрать некоторое количество камней из i -й кучки, чтобы из остальных кучек можно было выбрать некоторое количество камней, выполнив описанные условия.

К сожалению или к счастью, помощь прибыла слишком быстро, и Миша не успел найти ответ на свой вопрос. Поэтому сделать это предстоит вам.

Формат входных данных

Первая строка содержит два целых числа n и s ($1 \leq n \leq 100\,000$, $0 \leq s \leq 10^{18}$) — количество кучек с камнями, а также суммарное количество камней, взятых из кучек.

Каждая из следующих n строк содержит два целых числа l_i и r_i ($0 \leq l_i \leq r_i \leq 10^9$) — минимальное и максимальное количество камней, которые Миша может взять из i -й кучки.

Обратите внимание, что входные данные в этой задаче могут превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

Формат выходных данных

Выведите n целых чисел c_1, c_2, \dots, c_n , обозначающих количество способов выбрать некоторое количество камней из i -й кучки, чтобы существовала возможность взять некоторое количество камней из остальных кучек, чтобы выполнить поставленные условия.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Обозначим за m максимальную разность между r_i и l_i . Иными словами, $m = \max_{i=1}^n (r_i - l_i)$.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	10	$n, m \leq 7$		первая ошибка
2	15	$n, m \leq 1\,000$	1	первая ошибка
3	20	$n \leq 1\,000$	1, 2	первая ошибка
4	25	$n \cdot m \leq 10^7$	1, 2	первая ошибка
5	30	нет	1, 2, 3, 4	первая ошибка

Пример

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

Замечание

Рассмотрим пример из условия.

Из первой кучки можно выбрать 1, 2 или 3 камня. Для каждого из этих способов из остальных кучек можно выбрать некоторое количество камней таким образом, чтобы суммарно было выбрано 10 камней. Например, можно сделать это следующими способами: $1 + 2 + 3 + 3 + 1 = 10$, $2 + 2 + 3 + 2 + 1 = 10$, $3 + 2 + 3 + 1 + 1 = 10$.

Из второй кучки можно выбрать 2 или 3 камня, например, следующими способами: $1 + 2 + 3 + 3 + 1 = 10$, $1 + 3 + 3 + 2 + 1 = 10$.

Из третьей кучки можно выбрать 3 камня. Других способов нет, так как $l_3 = r_3 = 3$.

Из четвертой кучки можно выбрать 0, 1, 2 или 3 камня. Способы для выбора 1, 2 или 3 камней уже описаны выше. Способ для 0 камней выглядит следующим образом: $3 + 3 + 3 + 0 + 1 = 10$.

Из третьей кучки можно выбрать 1 камень. Других способов нет, так как $l_5 = r_5 = 1$.