

Задача А. Волк на стадионе

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

«Ну, погоди!» снова в деле! Чтобы в следующий раз уж точно догнать Зайца, Волк тренируется в беге с препятствиями на стадионе. Но мир не стоит на месте, и каждый день у какого-то препятствия может поменяться высота. Волк хочет знать после каждого изменения длину самого большого опасного отрезка на его дистанции, чтобы соответствующим образом морально подготовиться.

Дистанция из себя представляет n препятствий, расположенных на одной прямой. Препятствия для удобства занумерованы числами от 1 до n слева направо. Каждое препятствие характеризуется одним числом h_i — высотой (i -го) препятствия над землёй. Отрезок называется *опасным*, если на нем существует такое препятствие, что высоты препятствий левее него строго возрастают, а правее — строго убывают.

Помогите Волку найти длину самого большого опасного отрезка после каждого изменения. Гарантируется, что в любой момент времени нет двух рядом стоящих препятствий с одинаковой высотой.

Формат входных данных

В первой строке дано одно число n — количество препятствий на дистанции ($1 \leq n \leq 100\,000$). Во второй строке дано n чисел — высоты препятствий ($|h_i| \leq 10^{18}$).

В третьей строке дано число m — количество изменений ($1 \leq m \leq 100\,000$). В следующих m строках дано по два целых числа x и y — номер препятствия, высота которого изменилась, и новое значение высоты для этого препятствия, соответственно ($1 \leq x \leq n$, $|y| \leq 10^{18}$).

Формат выходных данных

Выведите m чисел, i -е из которых равно длине наибольшего опасного отрезка после i -го изменения.

Пример

стандартный ввод	стандартный вывод
9	6
1 2 3 4 5 4 3 2 1	6
5	4
3 10	5
2 5	5
7 100000000	
5 1	
3 1	

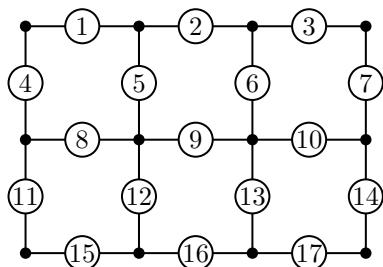
Задача В. Нырок Скруджа

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

Скрудж Мақдак просто обожает золото, и одно из любимых его занятий, конечно же, это нырять в золото!

Скрудж всегда ныряет в поле, имеющее вид сетки с тремя горизонтальными и четырьмя вертикальными рядами. По линиям сетки между её узлами можно перемещаться в обоих направлениях. В начале на каждой линии расположено несколько золотых монет.

Пример поля, куда ныряет Скрудж, приведен на рисунке, узлы сетки обозначены черными точками, линии — отрезками, а число на линии задает начальное количество золотых монет на ней.



Скрудж может нырнуть в любой узел сетки. Затем каждый раз Скрудж перемещается с узла, на котором он находится, на соседний узел по одной из линий сетки, на которой еще лежит хотя бы одна монета.

Проплывая вдоль линии, Скрудж забирает округленную вверх половину лежащих на ней монет. Таким образом, если на линии лежит x монет, то, пройдя по этой линии, Скрудж заберет $\lceil x/2 \rceil$ монет, а на линии останется $\lfloor x/2 \rfloor$ монет. По линиям, на которых монет уже нет, Скрудж перемещаться категорически отказывается.

Когда Скрудж оказывается в ситуации, что на всех соседних линиях сетки не осталось ни одной монеты, он заканчивает свой заплыв. После этого Скрудж считает число монет, которые он собрал.

Помогите Скруджу понять, какое максимальное количество монет он сможет собрать за один свой заплыв.

Формат входных данных

Входной файл состоит из пяти строк. Первая, третья и пятая строки содержат по три целых числа и описывают соответствующие горизонтальные линии. Вторая и четвертая строки содержат по четыре целых числа и описывают соответствующие вертикальные линии. Все числа неотрицательные и не превосходят 10^9 .

Формат выходных данных

В единственной строке выходного файла выведите число s — максимальное количество монет, которое Скруджу удастся собрать.

Примеры

стандартный ввод	стандартный вывод
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	150
1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1	7

Задача С. Модернизация дерева

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

В густом тропическом лесу живёт сообщество горилл. Их дом — огромное древнее дерево с множеством ветвей. Каждая ветвь соединяет два узла (развилки), и всего в дереве n узлов и $n - 1$ ветвей. Причём от любого узла можно добраться до любого другого по ветвям.

За последние годы дерево немного обветшало, и старейшины решили провести ремонт — укрепить самые важные ветви. Для этого они собрали совет из m уважаемых горилл-экспертов, чтобы каждый предложил, какие участки дерева нужно укрепить в первую очередь.

Каждый эксперт выбрал набор ключевых узлов a_1, \dots, a_s и считает, что необходимо укрепить все ветви, лежащие на пути между любыми двумя узлами из его списка (т. е. между a_i и a_j для всех $1 \leq i < j \leq s$).

Вы получили все списки от экспертов. Теперь вам нужно определить, какие ветви следует укрепить в первую очередь. Старейшины решили, что приоритет получат те ветви, которые упомянули хотя бы k экспертов. Ваша задача — подготовить список таких ветвей.

Формат входных данных

В первой строке входных данных даны числа n , m и k ($2 \leq n \leq 100\,000$, $1 \leq k \leq m \leq 50\,000$) — количество узлов в дереве, количество экспертов и число k соответственно.

В следующих $n - 1$ строках содержится описание дерева; в i -й из этих строк заданы два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), означающие что i -я ветвь в дереве проходит между узлами a_i и b_i .

В следующих m строках заданы ключевые узлы, выбранные каждым экспертом. i -я из этих строк начинается с целого числа s_i ($2 \leq s_i \leq n$), которое задаёт количество ключевых узлов, выбранных i -м экспертом. После него следуют s_i целых чисел, задающих эти узлы.

Гарантируется, что суммарная длина всех списков от экспертов не превосходит 100 000, то есть $\sum_{i=1}^m s_i \leq 100\,000$.

Формат выходных данных

В первой строке выходных данных выведите одно целое число r — количество ветвей, которые хотят укрепить по крайней мере k экспертов.

Во второй строке выведите r номеров этих ветвей в возрастающем порядке.

Пример

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

Задача D. Подотрезки-анаграммы

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

По данным двум массивам найдите максимальное число k , такое что в первом и втором массивах существуют подотрезки длиной k , совпадающие как анаграммы. Анаграммами назовём два слова (последовательности из чисел), в которых можно как-то переставить буквы (числа), чтобы они стали одинаковыми.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 1\,000$) — длина первого массива.

Во второй строке через пробел заданы n чисел a_i ($1 \leq a_i \leq 100\,000$) — первый массив.

В третьей строке дано число m ($1 \leq m \leq 1\,000$) — длина второго массива.

В четвертой строке через пробел заданы m чисел b_i ($1 \leq b_i \leq 100\,000$) — второй массив.

Формат выходных данных

В единственной строке выведите максимальная длина подотрезков, совпадающих как анаграммы.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 3 3 2 1	3
3 1 2 3 3 4 5 6	0

Задача Е. Рождество в Ктограде

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

В Ктоград приходит рождество, и подобревший Гринч готовится разносить подарки всю ночь.

Ктоград представляет из себя n домов, некоторые из которых соединены улицами, причем так, что между любыми двумя домами есть ровно один путь.

Гринч уже заготовил m мешков с подарками, но есть один нюанс. Ему самому нужно найти дом для празднования рождества (в пещере не так уютно). Раздачу подарков же он доверит своим верным помощникам. Одному помощнику можно дать ровно один мешок и отправить его в путь. При этом каждый помощник не ходит по одной и той же улице дважды. i -й мешок с подарками предназначен для жителей всех домов на пути от a_i -го дома до b_i -го. Поэтому считается, что помощнику можно дать i -й мешок с подарками, если он сможет выйти из места празднования Гринча и пройти через все дома на пути от a_i до b_i , при этом не проходя через одну и ту же улицу дважды.

Помогите Гринчу найти дом для празднования так, чтобы он смог отправить как можно больше мешков с подарками. Рождество — очень важный праздник, поэтому в вашем распоряжении любое число помощников.

Формат входных данных

В первой строке задано число n — количество домов в Ктограде ($2 \leq n \leq 2 \cdot 10^5$).

В следующих $n - 1$ строках описаны улицы. Улица задаётся числами x_i и y_i — номерами домов, которые она соединяет ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$). Гарантируется, что между любыми двумя домами существует единственный путь.

В следующей строке задано число m — количество мешков у Гринча ($1 \leq m \leq 2 \cdot 10^5$).

В следующих m строках описаны мешки с подарками. В i -й из них заданы числа a_i и b_i — начало и конец i -го пути ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$). Пути могут пересекаться и совпадать.

Дома нумеруются с единицы.

Формат выходных данных

Выполните одно число — максимальное число мешков с подарками, которые Гринч сможет раздать, если выберет оптимальный дом для празднования.

Пример

стандартный ввод	стандартный вывод
7	
1 2	
2 3	
3 4	
3 5	
5 6	
5 7	
3	
1 5	
2 4	
6 7	2

Задача F. Старая клавиатура

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

Вы обнаружили старую клавиатуру. Вы быстро обнаружили одну её неисправность: в процессе ввода сообщения на ней иногда внезапно срабатывала клавиша «Home», в результате чего курсор ввода перемещался в начало строки. Так, например, если в процессе ввода строки «abadefgcaba» клавиша «Home» сработала после ввода букв «d» и «g», то получится строка «cabaeafgabad». Ради интереса вы решили набрать строку s на ней, нажимая по очереди на соответствующие клавиши. Закончив набор, вы посмотрели на экран и увидел строку t . Теперь вы хотите понять, может ли она быть результатом вашего ввода, если единственная неисправность клавиатуры — лишние срабатывания клавиши «Home», либо в ней есть и другие проблемы.

Формат входных данных

В первой строке задано число n — длина строк s и t ($1 \leq n \leq 5000$). Во второй строке задана последовательность маленьких латинских букв длины n — строка s . В третьей строке задана последовательность маленьких латинских букв длины n — строка t .

Формат выходных данных

Выведите «YES», если из строки s могла получиться строка t , иначе выведите «NO».

Пример

стандартный ввод	стандартный вывод
11 irailikeyou ouilikeyira	YES

Задача G. Странная функция

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

Определим функцию $f(b, n)$ от двух целых чисел следующим образом:

- Если $n < b$, то $f(b, n) = n$
- Иначе $f(b, n) = f(b, \lfloor n/b \rfloor) + (n \bmod b)$

Здесь $\lfloor n/b \rfloor$ означает максимальное целое число не превышающее n/b , а $n \bmod b$ означает остаток от деления n на b .

Заметим, что $f(n, b)$ — сумма цифр числа n в b -ичной системе счисления.

Так например:

- $f(10, 87654) = 8 + 7 + 6 + 5 + 4 = 30$
- $f(100, 87654) = 8 + 76 + 54 = 138$

По двум заданным числам n и s вычислите минимальное такое $b \geq 2$, что $f(b, n) = s$, если оно существует. Выведите -1 , если такого b не существует.

Формат входных данных

Первая строка входных данных содержит целое число n ($1 \leq n \leq 10^{11}$) Вторая строка входных данных содержит целое число s ($1 \leq s \leq 10^{11}$)

Формат выходных данных

В единственной строке выходных данных выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 31415926535	-1
87654 30	10
87654 138	100
87654 45678	-1
31415926535 1	31415926535

Задача Н. Простая задача

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

Сегодня Марья Ивановна написала на доске первое простое число, после чего справа приписала к нему второе, затем третье и так далее. Всего Марья Ивановна выписала на доску первые n простых чисел. В результате её действий на доске появилось одно длинное число, которое начинается так: «23571113171923...». Напомним, что натуральное число называется простым, если у него есть ровно два различных натуральных делителя — единица и оно само.

Теперь Марья Ивановна задала классу следующую задачу: вычеркнуть из написанного на доске числа k цифр так, чтобы оставшееся на доске число было максимальным.

Помогите классу решить предложенную Марьей Ивановной задачу, а то после Нового года голова у них совсем не работает.

Формат входных данных

Входной файл к этой задаче содержит несколько наборов тестовых данных. В первой строке входного файла задано число T — количество наборов в файле.

В следующих T строках идут описания наборов, каждое из которых состоит из двух целых положительных чисел n и k . Гарантируется, что первые n простых чисел содержат в себе хотя бы $k + 1$ цифру суммарно.

Сумма всех n во входном файле не превосходит 400 000.

Формат выходных данных

Для каждого из тестовых наборов в отдельной строке выведите искомое максимальное число для соответствующих n и k .

Пример

стандартный ввод	стандартный вывод
2	57
4 2	711
5 3	

Замечание

В первом teste Марья Ивановна выписала число 2357. Максимальное число, которое может получиться после вычеркивания из него двух цифр: 57.

Во втором teste Марья Ивановна выписала число 235711. Максимальное число, которое может получиться после вычеркивания из него трех цифр: 711.

Задача I. Незапятнанная история

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

Михаил Павлович наконец-то завершил свою диссертацию. Учёному лишь осталось распечатать написанный на компьютере текст на n прямоугольных листах бумаги размера $h \times w$ сантиметров. После того, как диссертация была распечатана, Михаил Павлович сложил все листы бумаги в аккуратную стопку: наверху стопки лежала первая страница текста, под ней — вторая, и так далее.

Радости учёного не было предела до тех пор, пока он не решил отметить окончание работы чашкой кофе, и, конечно же, разлил весь напиток прямиком на стопку бумаги. В результате этого на каждом листе бумаги образовалось пятно, имеющее по счастливой случайности форму выпуклого многоугольника. Пятно было одинаково хорошо заметно с каждой стороны каждого листа бумаги. Более того, так как кофе медленно распространялся по стопке сверху вниз, пятно, образовавшееся на i -м сверху листе, оказалось *строго внутри* пятна, образовавшегося на $(i - 1)$ -м сверху листе.

Будем говорить, что многоугольник A *строго внутри* многоугольника B , если ломаные, образующие границы многоугольников, не имеют ни одной общей точки, а также любая вершина многоугольника A находится внутри многоугольника B .

Михаил Павлович сильно разозлился. От досады он схватил все n листов бумаги и яростно разбросал их по комнате. Немного успокоившись и выпив валерьянки, учёный решил собрать все листы обратно в стопку, но оказалось, что он забыл проставить номера страниц на листах, и теперь не знает, в каком порядке они шли изначально. Более того, некоторые листы во время полета по комнате неоднократно перевернулись.

Каждый лист бумаги при падении на пол мог либо оказаться в исходном состоянии, либо повернуться на 180 градусов таким образом, что текст теперь напечатан на листе не сверху вниз, а наоборот, либо перевернуться наизнанку, либо и перевернуться наизнанку, и повернуться на 180 градусов одновременно.

Михаил Павлович тут же решил посчитать, сколько существует различных способов собрать стопку бумаги таким образом, чтобы пятно на i -м сверху листе оказалось строго внутри пятна на $(i - 1)$ -м сверху листе. Перед тем, как собрать стопку, Михаил Павлович может повернуть и/или перевернуть каждый лист бумаги так, чтобы он оказался в одном из четырех описанных выше состояний.

Два способа собрать стопку считаются различными, если выполнено одно из двух условий:

1. Порядок следования листов в двух способах различается.
2. Порядок следования листов в двух способах совпадает, но есть хотя бы один лист, который находится в разных состояниях в первом и втором способе.

Формат входных данных

Первая строка содержит три целых числа n , w и h ($1 \leq n \leq 10^5$, $1 \leq w < h \leq 10^9$) — количество листов бумаги в диссертации и их размеры, соответственно.

Следующие строки содержат описания пятен, образовавшихся на каждом из n листов бумаги.

Первая строка описания каждого пятна содержит целое число k_i ($3 \leq k_i \leq 10^5$) — количество вершин в многоугольнике, задающем пятно, образовавшееся на i -м листе бумаги.

Каждая из следующих k_i строк содержит два целых числа x_{ij} и y_{ij} ($0 \leq x_{ij} \leq w$, $0 \leq y_{ij} \leq h$) — координаты вершин очередного многоугольника в порядке обхода против часовой стрелки. Будем считать, что начало координат каждого листа бумаги расположено в его левом нижнем углу.

Гарантируется, что все многоугольники являются выпуклыми, и никакие три вершины одного многоугольника не лежат на одной прямой.

Также гарантируется, что суммарное количество вершин всех многоугольников не превосходит 10^5 , то есть $\sum_{i=1}^n k_i \leq 10^5$.

Формат выходных данных

Выполните одно целое число — количество различных способов собрать стопку бумаги таким образом, чтобы пятно на i -м сверху листе оказалось строго вложено в пятно на $(i - 1)$ -м сверху листе.

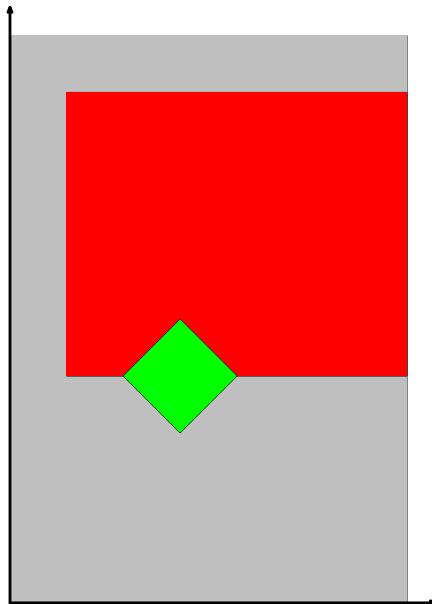
Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Пример

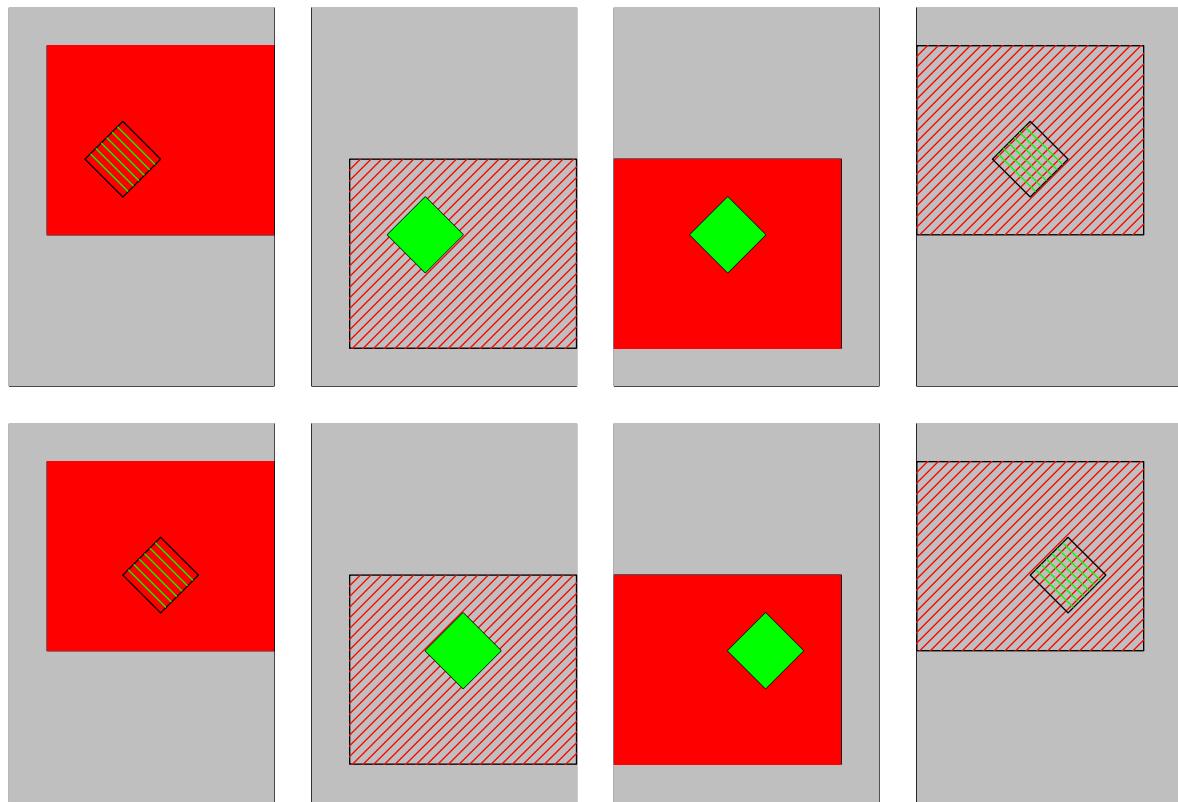
стандартный ввод	стандартный вывод
2 7 10 4 1 4 7 4 7 9 1 9 4 3 5 2 4 3 3 4 4	8

Замечание

Рассмотрим первый пример. На рисунке ниже изображены два многоугольника в их изначальных состояниях. Для удобства восприятия многоугольники наложены друг на друга. Первый многоугольник обозначен красным цветом, а второй — зеленым цветом. Лист бумаги обозначен серым цветом.



На рисунке ниже изображены восемь способов сложить стопку из данных двух листов. Если какой-то из многоугольников заштрихован, то это означает, что лист с данным многоугольником перевернут наизнанку.



Задача J. Ох уж эти туристы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Безаботная жаба-турист скакала по полянке, как вдруг, по незнанию местности, упала в колодец глубиной n метров. Теперь жаба лежит на самом дне колодца и ей предстоит долгий путь наверх.

Стенки колодца в некоторых местах скользкие, а в некоторых, наоборот, очень удобные. А именно, если жаба-турист сейчас находится на глубине x метров от уровня земли, то она может прыгнуть на любое расстояние от 0 до a_x метров вверх.

Жаба-турист путешествует не первый день, поэтому после каждого прыжка ей нужно отдохнуть. Если она отдохнет на глубине x метров от уровня земли, то за это время проскользит вниз на b_x метров.

Определите, за какое минимальное число прыжков жаба-турист может подняться до уровня земли, или же она обречена провести остаток своей жизни на дне колодца.

Формат входных данных

В первой строке задано целое положительное число n ($1 \leq n \leq 300\,000$) — глубина колодца.

Во второй строке задано n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq i$) — максимальная высота, доступная для прыжка с каждой глубины.

В третьей строке вводится n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq n - i$) — глубина, на которую жаба-турист провалится, взяв перерыв на каждой из глубин.

Формат выходных данных

В первой строке выведите целое число k — минимально возможное количество прыжков. В случае, если жаба-турист не может выбраться из колодца, выведите -1 .

Если жаба-турист может выбраться из колодца, то во второй строке выведите последовательность глубин длины k , на которые она будет запрыгивать каждым своим прыжком. Считайте глубину вне колодца в любой точке равной нулю.

Если существует несколько решений, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
3 0 2 2 1 1 0	2 1 0
2 1 1 1 0	-1
10 0 1 2 3 5 5 6 7 8 5 9 8 7 1 5 4 3 2 0 0	3 9 4 0

Замечание

В первом тесте из условия жаба находится на дне колодца, за один прыжок вверх поднимается к отметке в 1 метр глубины. После этого она соскальзывает вниз на метр и оказывается на отметке в 2 метра. С этой отметки она уже сможет выпрыгнуть из колодца за один прыжок.

Во втором тесте из условия жаба может допрыгнуть до отметки в один метр, но сразу после этого соскользнет обратно на дно колодца, поэтому ей не выбраться.

В третьем тесте из условия выбраться из колодца можно только прыгнув с глубины 5. Попасть напрямую туда нельзя, но можно добраться с помощью серии прыжков $10 \rightarrow 9 \rightarrow 4 \dashrightarrow 5$, где прыжок вверх обозначается простой стрелочкой, а пунктиром обозначен спуск

Задача К. Мойша в банке

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

Мойша хочет положить в один банк n мешков с деньгами (говорят, в том банке самые выгодные условия). Мойша очень осторожный и влиятельный, поэтому у него бесконечно много бронемашин для перевозки денег, в каждую машину вмещается k мешков. В i -м мешке c_i шекелей.

Мешки расположены подряд, когда Мойша будет складывать их, он положит первые k мешков в первую бронемашину, следующие k мешков во вторую бронемашину и т.д.

Прежде чем складывать мешки Мойша хочет поменять их порядок так, чтобы сумма шекелей в первой бронемашине была максимальна, сумма шекелей во второй бронемашине была максимальна среди оставшихся мешков и т.д.

За одно действие Мойша может выбрать любой мешок и поставить в любое место в последовательности.

Скажите, какое минимальное кол-во действий надо сделать Мойше, чтобы увезти все деньги.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq n, k \leq 10^5$) — количество мешков и вместительность бронемашин.

Во второй строке по n целых чисел c_i ($1 \leq c_i \leq 10^5$) — количество шекелей в каждом мешке.

Формат выходных данных

Скажите какое минимальное кол-во действий надо сделать Мойше, чтобы увезти все деньги.

Примеры

стандартный ввод	стандартный вывод
6 2 10 4 5 3 11 100	3
5 2 100 78 72 95 2	1
6 2 72 6 52 52 99 53	3
7 2 78 29 5 93 31 10 29	2

Задача L. Копирка

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

Работа с большими наборами чисел требует быстро копировать данные из одного массива в другой. Конечно же, вы заинтересованы в том, чтобы выполнять такие операции как можно быстрее.

Вам даны два массива целых чисел одинаковой длины n : a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n . Также нужно обрабатывать m запросов двух типов:

1. Присвоить подотрезок массива a длины k , начиная с позиции p , в массив b , начиная с позиции q , то есть выполнить $b_{q+j} = a_{p+j}$ для всех целых j ($0 \leq j < k$). Гарантируется, что запрос корректен: оба подотрезка полностью находятся внутри границ соответствующих массивов.
2. Вывести значение элемента в позиции p массива b , то есть вывести b_p .

Формат входных данных

В первой строке через пробел заданы два целых числа n и m ($1 \leq n, m \leq 10^5$) — количество элементов в массивах и число запросов. Вторая строка содержит массив a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$). Третья строка содержит массив b_1, b_2, \dots, b_n ($|b_i| \leq 10^9$).

Далее следуют m строк с описанием запросов. В i -й строке сначала указано целое число t_i — тип i -го запроса ($1 \leq t_i \leq 2$). Если $t_i = 1$, то далее идут три целых числа p_i, q_i, k_i ($1 \leq p_i, q_i, k_i \leq n$) — параметры операции копирования. Если $t_i = 2$, то далее записано одно число p_i ($1 \leq p_i \leq n$) — позиция в массиве b .

Гарантируется корректность всех запросов, то есть операции не выходят за пределы массивов a и b .

Формат выходных данных

Для каждого запроса второго типа выведите одно число в отдельной строке — результат операции.

Пример

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