

Задача А. Did he drop any good loot?

Слава играет в очень крутую RPG. Он зачистил подземелье, выбил n крутых артефактов и теперь хочет как можно выгоднее их продать. Для каждого артефакта Слава знает его цену p_i и массу w_i . Слава не может унести на продажу артефакты с суммарной массой более m . Но Слава может активировать некоторые артефакты. При активации i -ый артефакт увеличивает Слаvinу силу и прибавляет к пределу переносимой массы d_i (активированные артефакты не исчезают, их стоимость и масса не изменяются, их также можно потом продать). Одновременно Слава может иметь активированными не более 2 артефактов.

Более формально, Слава хочет выбрать множество артефактов с максимальной суммарной стоимостью среди всех корректных множеств. Множество артефактов $i_1, i_2, \dots, i_k$ ($k \leq n$) называется корректным, если $w_{i_1} + w_{i_2} + \dots + w_{i_k} \leq m + s$, где s — эффект от активации артефактов: $s = d_{j_1} + d_{j_2}$, если Слава активировал артефакты j_1 и j_2 ($j_1 \neq j_2$), $s = d_j$ — если он активировал только артефакт j , и $s = 0$, если Слава не активировал артефакты (j_1, j_2 и j обязаны входить в множество взятых артефактов $i_1, i_2, \dots, i_k$).

Формат входных данных

В первой строке содержатся два целых числа через пробел: n и m ($1 \leq n \leq 10^4$, $1 \leq m \leq 500$) — количество артефактов и максимальная масса, которую способен унести Слава, соответственно.

Далее в n строках содержатся по три целых числа через пробел: p_i , w_i и d_i ($1 \leq p_i \leq 10^5$, $1 \leq w_i \leq 100$, $0 \leq d_i \leq 100$) — соответственно стоимость артефакта, его масса, а также прибавка к пределу переносимой массы, которую он даёт при активации.

Формат выходных данных

Выведите единственное целое число — максимальную суммарную стоимость унесённых Славой артефактов.

Примеры

стандартный ввод	стандартный вывод
5 10 1 5 3 2 4 0 3 2 2 4 1 4 5 3 1	15
3 10 100 100 20 200 80 30 300 60 40	0

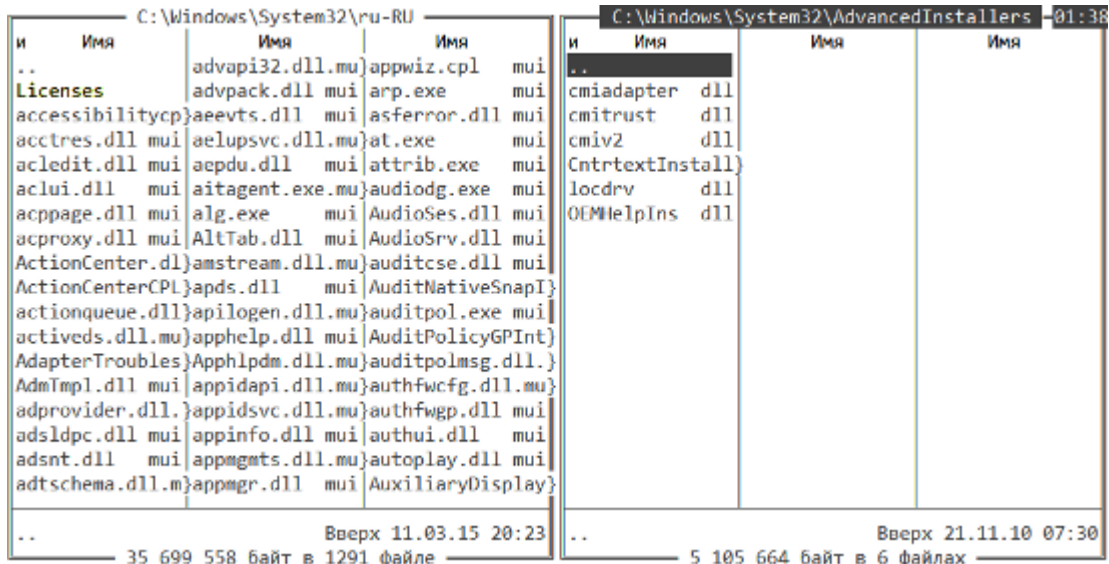
Замечание

В первом тесте Слава не может просто так взять все артефакты — он может унести только массу до 10, а суммарная масса артефактов — 15. Поэтому Слава активирует артефакты 1 и 4, после чего его предел переносимой массы становится равным $10 + 3 + 4 = 17$ (Слава также мог активировать любую пару артефактов, дающих в сумме прибавку 5 или более).

Во втором тесте Слава не может унести никакие артефакты — ни с активацией некоторых, ни без оной.

Задача В. Far Manager

Паша очень любит файловый менеджер Far. При настройках по умолчанию Far выстраивает файлы в несколько столбцов, каждый из которых вмещает в себя p файлов. Кроме того, «нулевым файлом» всегда является ссылка на предыдущую папку, а после этой ссылки идут непосредственно папки и файлы, находящиеся в данной папке.



и	Имя	Имя	Имя	и	Имя	Имя	Имя
..	advapi32.dll	advpack.dll	advapi32.dll	..	cmiadapter	dll	
Licenses	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
accessibilitycp	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
acctrres.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
acledit.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
aclui.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
acppage.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
acproxy.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
ActionCenter.dl	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
ActionCenterCPI	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
actionqueue.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
activeds.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
AdapterTroubles	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
AdmImpl.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
adprovider.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
adslpdc.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
adsnt.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
adtschema.dll	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		
..	advapi32.dll	advpack.dll	advapi32.dll	cmiadapter	dll		

Паша зашёл в папку, где находятся n файлов, последовательно занумерованных от 1 до n . Ему необходимо выделить файл с номером x . Для этого он должен поставить курсор на строчку с этим файлом. Он может нажимать одну из четырёх стрелок. Стрелки вниз и вверх смещают курсор на 1 позицию, а стрелки вправо и влево — на p позиций (или помещают курсор на последний/нулевой файл, если в направлении движения курсора осталось менее p файлов).

Помогите Паше определить, какое минимальное количество нажатий на клавиши ему потребуется, чтобы выделить файл с номером x . Изначально курсор указывает на ссылку на предыдущую папку.

Формат входных данных

В единственной строке записаны 3 целых числа p, n, x ($1 \leq p \leq 10^9, 1 \leq n \leq 10^9, 1 \leq x \leq n$) — максимальное количество файлов в столбце, количество файлов в папке, в которую зашёл Паша, и номер файла, который ему надо выделить.

Формат выходных данных

Выведите единственное целое число — минимальное количество нажатий на клавиши, которое необходимо Паше, чтобы выделить файл с номером x .

Примеры

стандартный ввод	стандартный вывод
5 8 3	3
6 20 14	4
8 33 33	5

Задача С. Современное искусство

Серёжа — современный художник, поэтому в своих композициях он использует 3D-принтер. В данный момент Серёжа придумал идею следующей композиции. Он напечатает n разноцветных пластиковых брусков и попытается расположить эти бруски горизонтально один над одним (бруски можно считать горизонтальными отрезками на плоскости, имеющими ненулевую длину и лежащими на попарно различных прямых, параллельных оси абсцисс), но сделать это следует с учетом соотношений цветов (Серёжа же всё-таки художник!). Соотношения бывают двух видов: соотношение предшествования (тип А) и соотношение пересечения (тип В).

Если для двух брусков i и j есть соотношение типа А, то брусок i должен лежать строго левее бруска j (то есть координата по оси абсцисс любой точки бруска i должна быть строго левее координаты по оси абсцисс любой точки бруска j).

Соотношение типа В между брусками i и j означает, что данные бруски должны иметь хотя бы по одной точке, не являющейся границей бруска, проекции которых на ось абсцисс совпадают.

Если между двумя брусками нет ни соотношения А, ни соотношения В, то они могут располагаться друг относительно друга в любом порядке.

Помогите Серёже найти идеальные координаты по оси абсцисс для своих брусков или определить, что это невозможно. В случае успеха своей композиции он, может быть, даже поделится с вами гонораром!

Формат входных данных

В первой строке задано три целых числа через пробел: n , a и b ($1 \leq n \leq 10^5$, $0 \leq a, b \leq 10^5$) — количество брусков, а также количество ограничений первого и второго типа соответственно.

В следующих a строках указаны пары целых чисел $f_a[i]$ и $s_a[i]$ ($1 \leq f_a[i], s_a[i] \leq n$, $f_a[i] \neq s_a[i]$) — номера брусков, для которых должно выполняться соотношение типа А.

В следующих b строках указаны пары целых чисел $f_b[i]$ и $s_b[i]$ ($1 \leq f_b[i], s_b[i] \leq n$, $f_b[i] \neq s_b[i]$) — номера брусков, для которых должно выполняться соотношение типа В.

Формат выходных данных

Если расположения брусков, удовлетворяющего данным требованиям, не существует, выведите в единственной строке «NO» (без кавычек).

В противном случае в первой строке выведите «YES» (без кавычек). Далее, для каждого из n брусков на отдельной строке выведите два целых числа l_i и r_i ($0 \leq l_i < r_i \leq 10^9$) — координаты левого и правого концов i -го бруска по оси абсцисс в удовлетворяющем всем требованиям расположении.

Примеры

стандартный ввод	стандартный вывод
4 2 1 1 2 2 3 3 4	YES 0 1 2 3 4 6 5 7
3 2 0 1 2 2 1	NO

Задача D. Нечестная игра

Игра в абстрактные n -гранные кости описывается следующими правилами. В игре участвуют два игрока и некоторое количество абстрактных n -гранных костей, каждая из граней которых соответствует целому числу, причём числа на всех гранях всех костей различны. В начале игры первый игрок выбирает себе кость, после чего второй игрок, зная выбор первого, выбирает свою кость (разумеется, он не может выбрать кость, уже взятую первым игроком). Затем игроки одновременно бросают свои кости, и тот игрок, на чьей кости выпало число больше, чем у соперника, побеждает.

Константин хочет сыграть бесконечное число партий в эту игру с Михаилом. Для этого ему требуется создать три n -гранных кости, обладающие следующим свойством: какую бы кость ни выбрал первый игрок, второй игрок всегда может выбрать свою кость так, что вероятность победы второго игрока будет строго больше $\frac{1}{2}$. Константин планирует уступить своему сопернику право первого хода: делая ход вторым, он будет побеждать чаще, выбирая нужную кость, и благодаря этому сможет выиграть бесконечное количество денег.

Михаил согласился на эти условия, однако за ним осталось право выбора числа n . Помогите Константину создать требуемый набор n -гранных костей.

Формат входных данных

В единственной строке записано единственное целое число n ($1 \leq n \leq 1000$) — количество граней на костях.

Формат выходных данных

Если требуемого набора абстрактных n -гранных костей не существует, выведите «FAIL».

Иначе в первой строке выведите «WIN», а в каждой из следующих трёх строк выведите по n целых чисел, которые должны быть указаны на соответствующих костях. Все $3n$ чисел должны быть различны и находиться в диапазоне от 1 до $3n$. Если существует несколько возможных решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
1	FAIL
5	WIN 1 3 7 14 15 2 6 10 11 13 4 5 8 9 12
6	WIN 4 6 8 9 11 18 2 3 7 13 15 17 1 5 10 12 14 16

Задача Е. СуперГиперМаркет

Вот и наступил день открытия самого большого магазина в мире. Для показа всех возможностей своего магазина, в том числе невероятной скорости обслуживания покупателей на кассе, владельцы придумали такой рекламный ход: в магазин запускается n самых средних покупателей мира. Кассы же откроются лишь тогда, когда все покупатели наберут себе вещей и встанут в очереди на оплату. Но если каждый будет ходить и выбирать самую маленькую очередь, то это затянется на несколько часов. Поэтому вас попросили автоматизировать выбор очереди для каждого подходящего покупателя.

Отдел социологов рассказал вам, как средний покупатель выбирает очередь: он смотрит на количество людей в ней и на количество покупок двух последних людей в данной очереди. Формально, средний покупатель выбирает очередь в ту кассу i , для которой минимален показатель $x_i = c_i \cdot p$, где c_i — количество покупателей в i -ой очереди, а p — среднее количество покупок у одного покупателя очереди. Число p определяется максимум по двум последним покупателям в очереди: $p = (p_1 + p_2)/2$, если в очереди 2 и больше людей, где p_1 и p_2 — количество покупок последнего и предпоследнего покупателя, $p = p_1$, если покупатель в очереди всего один, и $p = 0$, если покупателей в очереди нет. Из всех касс с минимальным x_i покупатель выбирает кассу с минимальным номером i .

После выбора кассы покупатель встает в очередь к данной кассе и никуда из нее не уходит.

Формат входных данных

В первой строке содержатся два целых числа через пробел: n и k ($1 \leq n, k \leq 10^5$) — количество потенциальных покупателей и количество касс в магазине.

В следующей строке содержатся n целых чисел через пробел: p_j ($1 \leq p_j \leq 1000$) — количество покупок у j -ого в хронологическом порядке покупателя. Гарантируется, что каждый покупатель успевает выбрать очередь и встать в неё до того, как кассу начинает выбирать следующий покупатель.

Формат выходных данных

В единственной строке выведите n чисел через пробел. i -е число должно быть равно номеру кассы, в очередь к которой стоит встать i -му покупателю, руководствуясь критерием, описанным в задаче.

Примеры

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

Задача F. Чемпионат Берляндки

Каждый год БерГУ (Берляндский Государственный Университет) проводит олимпиаду по программированию под названием «Чемпионат Берляндки». В этом году задачи помогал готовить Андрей — студент БГАУ (Берляндский Государственный Альтернативный Университет). Всего было подготовлено m задач.

Андрей очень гордится своим вузом, поэтому хочет, чтобы команда БГАУ выиграла. Он, конечно, не может сливать условия задач командам БГАУ до начала конкурса, однако он может дать совет, как сформировать команды.

Всего от БГАУ в олимпиаде собирается участвовать n студентов. Для каждого студента Андрей знает, какие из задач он способен решить, а также какова его производительность — максимальное количество задач, которое он может решить во время конкурса. Каждая команда должна состоять максимум из трёх студентов. Андрей верит в студентов БГАУ, поэтому считает, что каждая команда во время конкурса будет действовать оптимально, то есть решит максимально возможное для себя количество задач (при этом ни один из членов команды не решит больше задач, чем его производительность, а также команда не сможет решить задачи, которые не умеет решать ни один из её членов).

Помогите Андрею сформировать ровно одну команду из трёх студентов таким образом, чтобы она смогла решить наибольшее возможное количество задач среди всех возможных команд БГАУ.

Формат входных данных

В первой строке записано два целых числа n, m ($3 \leq n \leq 50, 1 \leq m \leq 1000$) — количество студентов БГАУ и количество задач соответственно.

Во второй строке записан массив p , состоящий из n целых чисел, где p_i ($0 \leq p_i \leq m$) — производительность i -го студента.

В i -ой из следующих n строк записана строка из заглавных и строчных латинских букв $name_i$ ($|name_i| \leq 10$) — имя i -го студента. Имена всех студентов попарно различны.

Далее записана бинарная матрица a размера $n \times m$, строки которой соответствуют студентам, а столбцы — задачам. Если $a_{i,j}$ равно 1, то i -й студент может решить j -ю задачу, иначе — не может.

Формат выходных данных

В первой строке выведите единственное целое число s — максимальное количество задач, которое может решить сформированная Андреем команда.

Во второй строке выведите через пробел имена студентов, составляющих эту команду. Имена студентов можно выводить в любом порядке.

Если существует несколько оптимальных ответов, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
5 5 3 2 2 3 1 Slava Andrew Egor Denis Sergey 11011 10100 00001 00100 10000	5 Slava Andrew Egor
4 6 1 2 1 2 Denis Andrew Egor Slava 100011 111100 000101 000111	5 Denis Andrew Slava

Задача G. Потерянный граф

Студент Владислав пришёл на экзамен по программированию, совершенно не подготовившись, и, как назло, ему достался билет про какой-то непонятный алгоритм на графе, который абсолютно точно никогда не пригодится ему в жизни. Недолго думая, он попросил у сидящей рядом студентки шпаргалку по этому билету и обнаружил в ней следующий псевдокод алгоритма:

функция пвг(в)

начало

 посещено[в] := истина

 печатай("in " + в)

 для к := 1 : н делай

 начало

 если являются_соседями(в, к) и не посещено[к] делай

 начало

 пвг(к)

 конец

 конец

 печатай("out " + в)

конец

Владислав вручную выполнил его на придуманном им самим связном неориентированном графе и получил некоторый результат. К сожалению, от волнения он съел бумажку, на которой был нарисован этот граф, а ведь он должен показать его преподавателю. Помогите Владиславу восстановить граф. Это нужно сделать как можно быстрее, поэтому в полученном графе должно быть минимально возможное количество рёбер.

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 10^5$) — количество вершин в графе.

Далее следует результат работы алгоритма, который никогда в жизни не пригодится Владиславу, — $2n$ строк вида «in x » или «out x », где x — номер вершины ($1 \leq x \leq n$).

Гарантируется, что все вершины от 1 до n присутствуют в графе, и что входные данные действительно задают результат работы алгоритма на некотором реально существующем графе.

Формат выходных данных

Выведите $(n - 1)$ строку, в каждой по 2 целых числа — номера вершин, соединённых очередным ребром. Выводить вершины в паре и пары вершин можно в любом порядке.

Примеры

стандартный ввод	стандартный вывод
2 in 1 in 2 out 2 out 1	1 2
4 in 4 in 2 out 2 in 3 in 1 out 1 out 3 out 4	1 3 2 4 3 4

Задача Н. Шумная лекция

Преподаватель N/A читает лекцию. Лекция состоит из n предложений и проходит так: каждое предложение надо сначала запомнить с тетрадки (на это уходит a_i времени), а затем написать на доске (на это уходит b_i времени). Если в процессе запоминания в аудитории стоит какой-то шум, то он отвлекает N/A, и после того, как шум закончился, запоминать предложение придётся заново. А если шум возникает в процессе написания на доске, то заново придётся запоминать ту часть предложения, которая ещё не написана (потратив на это соответствующую долю времени от a_i).

На лекции присутствуют k студентов, каждый из которых может ровно один раз, в произвольный момент времени, инициировать шум продолжительностью c_j . На какое максимальное количество времени удастся замедлить лекцию?

Формат входных данных

В первой строке записано единственное целое число n ($1 \leq n \leq 1000$) — количество предложений в лекции.

Во второй строке записано n целых чисел a_i ($1 \leq a_i \leq 1000$) через пробел — продолжительность запоминания i -го предложения.

В третьей строке записано n целых чисел b_i ($1 \leq b_i \leq 1000$) через пробел — длительность написания i -го предложения на доске.

В четвёртой строке записано единственное целое число k ($1 \leq k \leq 1000$) — количество студентов, присутствующих на лекции.

В пятой строке записано k целых чисел c_j ($1 \leq c_j \leq 1000$) через пробел — продолжительность шума, который может инициировать j -й студент.

Формат выходных данных

Выведите единственное число — максимальное количество времени, на которое удастся замедлить лекцию.

Примеры

стандартный ввод	стандартный вывод
3 3 2 1 1 5 3 2 1 2	9
4 2 1 2 1 1 2 3 4 3 2 3 1	12

Задача I. Пусти козу в поле

Коза Таня случайно выбралась из своего огорода и оказалась в поле. Коза сильно растерялась, так как ни разу в своей жизни здесь не была, поэтому решила идти прямо в одном направлении. Поле состоит из бесконечного количества клеток размера 1×1 , каждая клетка соединена с четырьмя другими через общие стороны. За один ход коза перемещается в соседнюю по направлению своего движения клетку.

Пастухи заметили пропажу и пытаются поймать козу. Каждый пастух за один ход может переместиться в любом направлении на одну или две клетки, однако он не может менять направление движения в течение хода. Пастух также может не двигаться во время своего хода, то есть остаться в той же клетке. Считается, что пастух поймал козу, если на момент окончания своего хода он находится с ней в одной клетке. В начальный момент времени ни один из пастухов не находится в одной клетке с козой.

Таня и пастухи ходят по очереди — сначала коза, затем пастухи в том порядке, в котором они перечислены во входных данных. Определите, какой пастух первым поймает козу.

Формат входных данных

В первой строке записаны два целых числа x, y ($|x|, |y| \leq 1000$) — начальные координаты козы.

Во второй строке записана одна строка s , определяющая направление движения козы. s может принимать 4 значения (кавычки даны для наглядности):

- «LEFT» — коза движется влево, координата x уменьшается на единицу,
- «RIGHT» — коза движется вправо, координата x увеличивается на единицу,
- «UP» — коза движется вверх, координата y увеличивается на единицу,
- «DOWN» — коза движется вниз, координата y уменьшается на единицу.

В третьей строке дано одно число n ($1 \leq n \leq 1000$) — количество пастухов.

В i -й из n последующих строк заданы непустая строка name_i ($|\text{name}_i| \leq 20$), состоящая из строчных латинских букв, и два целых числа x_i, y_i ($|x_i|, |y_i| \leq 1000$) — имя i -го пастуха и его начальные координаты. Гарантируется, что не существует двух пастухов с одинаковым именем.

Формат выходных данных

Выведите единственную строку — имя пастуха, который первым поймает козу.

Примеры

стандартный ввод	стандартный вывод
0 0 LEFT 3 andrew -10 0 denis 10 0 ilia 0 10	andrew
10 2 UP 3 danila 5 4 sashac 11 1 sashab 12 10	sashac
0 0 UP 2 mike 2 1 constantine 0 1	mike

Задача J. Покупка билетов

Скоро в городе Н-ске пройдет крупнейшее студенческое соревнование по ACM (Anarchy Code Modification). Слава является старшим в группе студентов своего вуза, участвующих в этом соревновании, поэтому ему необходимо купить билеты на поезд на всю группу. Каждый студент указал номер места, на котором он хотел бы ехать (все студенты назвали разные номера).

Кассирши вокзала объявили забастовку, и Славе пришлось воспользоваться терминалом. Покупка билетов в терминале производится следующим образом:

1. Указывается информация о людях, на которых покупаются билеты (за одно обращение к терминалу можно указать не более k человек).
2. Указываются нижняя и верхняя граница номеров мест включительно, на которые следует выписать билеты. В этих границах могут быть и занятые места. Билеты выписываются на s первых свободных мест выделенного отрезка, где s — количество людей в текущем обращении к терминалу. Если в отрезке недостаточно свободных мест, то покупка не производится.

Каждое обращение к терминалу сопровождается вводом большого количества дополнительной информации, поэтому Слава хотел бы минимизировать количество обращений. Сам Слава уже не успевает определить оптимальный порядок покупок, иначе он даже бегом не догонит последний автобус домой, поэтому просит вас помочь ему.

Формат входных данных

В первой строке содержатся три целых числа через пробел: n , m и k ($1 \leq n, m, k \leq 10^5$, $n \leq m$) — количество студентов в группе, количество свободных мест в вагоне и максимальное количество людей в одном обращении к терминалу соответственно.

Во второй строке содержится n различных целых чисел w_i , упорядоченных по возрастанию ($1 \leq w_i \leq 10^9$) — номер места, на котором хочет ехать i -й студент.

В третьей строке содержатся m различных целых чисел f_j , упорядоченных по возрастанию ($1 \leq f_j \leq 10^9$) — номера свободных мест в вагоне.

Гарантируется, что все студенты желают ехать только на свободных местах, т.е. для любого $1 \leq i \leq n$ существует такое $1 \leq j \leq m$, что $w_i = f_j$.

Формат выходных данных

В первой строке выведите единственное целое число s — минимальное количество обращений к терминалу.

В следующих s строках выведите информацию об обращениях к терминалу — количество людей в обращении и их номера через пробел. Каждый человек должен быть указан только в одном обращении, билеты должны быть куплены на всех студентов группы.

Примеры

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

стандартный ввод
12 21 4 2 6 8 10 12 28 40 44 46 48 50 52 2 4 6 8 10 12 24 26 28 30 32 33 34 35 36 40 44 46 48 50 52
стандартный вывод
5 1 1 4 2 3 4 5 1 6 4 7 8 9 10 2 11 12

Задача К. Рейтинг команды

Лёша и Костя создали свой Codeforces. Чтобы привлечь внимание, они решили провести соревнование Jaguar Turnik Cup. Также, как и в VK Cup, в нём могут участвовать команды из двух человек, но Лёша пошел дальше и запретил участвовать в соревновании командам из одного человека. Рейтинг каждого участника они также взяли с Codeforces. Осталось только придумать, как высчитывать рейтинг команды.

После долгих споров было решено использовать следующую формулу: $\text{rating}_{\text{team}} = x \cdot \text{rating}_1 + y \cdot \text{rating}_2$, где rating_1 — максимальный из рейтингов членов команды, а rating_2 — минимальный. Осталось только подобрать коэффициенты x и y .

После ещё одного раунда споров все согласились, что они должны удовлетворять следующим условиям: $x + y = 1$, $x, y \geq 0$. Однако о точных значениях договориться не удалось, поэтому они пригласили своих друзей поучаствовать в разминочном раунде. Всего было составлено n команд, по 2 человека в каждой, для каждого участника был известен его рейтинг на Codeforces.

Раунд прошёл успешно, и создатели решили на основе результатов раунда выбрать такие коэффициенты x и y , чтобы команды в таблице результатов были отсортированы в порядке невозрастания рейтингов. Удастся ли им это сделать?

Формат входных данных

В первой строке дано единственное целое число n ($1 \leq n \leq 10^6$) — количество команд, участвовавших в разминочном раунде.

В i -ой из последующих n строк даны два числа a_i, b_i ($1 \leq a_i, b_i \leq 10000$) — рейтинги первого и второго членов команды, занявшей i -е место.

Формат выходных данных

Выведите «YES» или «NO» (без кавычек) — в зависимости от того, возможно ли подобрать коэффициенты требуемым образом.

Примеры

стандартный ввод	стандартный вывод
3 2106 1946 1955 1859 1778 1669	YES
3 10 2 8 8 9 6	NO

Задача L. Ханойские башни и прогресс

Андрей очень не любит Ханой за его консервативность — уже сотни лет самыми высокими зданиями там считаются три древние башни, между которыми каждый год переносят гигантские кольца по одному и тому же алгоритму (см. замечание). Каждый год Андрея мучает осознание того, что гигантские деньги тратятся на какое-то непонятное развлечение — ещё бы, двигают эти кольца с помощью кучи подъемных кранов, перекрывая центр города на несколько дней! Но прогресс не стоит на месте, и уже в следующем году в Ханое будут построены дополнительно k башен поменьше, на каждой из которых можно будет поместить не более одного кольца одновременно. Когда Андрей прочитал об этом, его тут же заинтересовало, какое минимальное количество действий переноса для n колец потребуется после этого нововведения. Сам он не хочет тратить ни секунды времени на этот ненавистный ему город, поэтому просит вас помочь произвести этот несложный подсчёт.

Формат входных данных

В единственной строке содержатся два целых числа через пробел: n и k ($1 \leq n \leq 10^9$, $0 \leq k \leq 10^9$) — количество колец и количество дополнительных башен соответственно (общее количество башен будет равно $k + 3$).

Формат выходных данных

Выведите минимальное количество действий, необходимое для перемещения всех колец с первой башни на третью. Так как количество действий может быть слишком велико, выведите остаток от деления этого числа на $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 0	7
3 1	5
42 0	46480317
1000 0	688423209

Замечание

«Ханойские башни» — головоломка, в которой даны три стержня («башни»); на один из стержней («стартовый») нанизано некоторое количество колец, причем все кольца разного размера и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из колец за наименьшее число ходов на другой («конечный») стержень. За один раз разрешается переносить на один из стержней только одно верхнее кольцо, причём нельзя класть большее кольцо на меньшее.

Задача М. Пятый сезон

Серсея и Джейме украли из запасов Бобрового Утёса n золотых монет. Пока Тайвин не заметил пропажи, они решили сыграть в следующую игру.

Игра состоит из q раундов. В начале каждого раунда все n монет лежат в одной кучке. Игроки по очереди берут из кучки не менее k_i и не более l_i монет ($k_i \leq l_i$, i — номер раунда). Проигрывает тот игрок, который не может сделать ход.

Серсея и Джейме определили для каждого из q раундов пары чисел (k_i, l_i) . Поскольку того, кто ходит первым, Серсея и Джейме определяют подбрасыванием монетки, они не видят большого смысла выяснять, кто из них в каком раунде победит. Им интересна максимально возможная длительность каждого раунда при условии, что каждый из них играет оптимально, то есть стремится выиграть. Длительность раунда определяется суммарным количеством ходов Серсеи и Джейме.

Ваша задача — определить максимально возможную длительность каждого раунда игры.

Формат входных данных

В первой строке содержатся целые числа n и q — количество монет и количество раундов соответственно ($1 \leq n \leq 10^6, 1 \leq q \leq 10^5$).

В каждой из следующих q строк содержится пара целых чисел k_i и l_i ($1 \leq k_i \leq l_i \leq n$, $i = 1, 2, \dots, q$) — минимальное и максимальное количество монет, которое игрок может взять за один ход в i -м раунде.

Формат выходных данных

Выведите q целых чисел через пробел. Число на позиции i должно быть равно максимально возможной длительности i -го раунда.

Примеры

стандартный ввод	стандартный вывод
10 2 1 1 5 6	10 1