

Задача А. Красота фейерверка

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

В лаборатории теоретической пиротехники изучают новые технологии организации фейерверков. Фейерверк представляется как корневое дерево, а поскольку в мощном фейерверке его элементы также взрываются, порождая новые фейерверки, то ученые вводят операцию возведения корневого дерева в степень.

Корневое дерево содержит одну или несколько вершин. Одна из вершин выделена и называется *корнем* дерева, для каждой из остальных вершин ровно одна другая вершина является *родителем*. При этом от любой вершины можно добраться до корня, последовательно переходя от вершины к ее родителю. Вершина, которая не является родителем никакой другой вершины, называется *листом*. Если вершина x является родителем вершины y , то вершина y является *ребенком* вершины x . Будем говорить, что вершина и ее родитель *соединены ребром*.

На рис. 1 показан пример корневого дерева с корнем в вершине 1. Родителем вершин 2 и 3 является вершина 1, родителем вершины 4 является вершина 2. Вершины 2 и 3 — дети вершины 1, а вершина 4 — ребенок вершины 2. Листьями являются вершины 3 и 4.

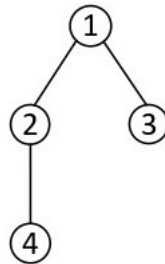


Рис. 1. Пример корневого дерева с корнем в вершине 1, листьями 3 и 4.

Фейерверк задается своим *базовым деревом* T и *мощностью* m . Фейерверк представляется деревом, которое получается в результате возведения дерева T в степень m . Операция возведения дерева в степень устроена следующим образом. Если $m = 1$, то результат T^1 — само дерево T . Для $m > 1$ рассмотрим дерево T^{m-1} . Выполним следующую операцию: для каждого листа x дерева T^{m-1} создадим копию дерева T и назовем лист x родителем корня соответствующей копии. Получившееся дерево будет деревом T^m .

На рис. 2 показано дерево, представленное на рис. 1, в степенях 1, 2 и 3.

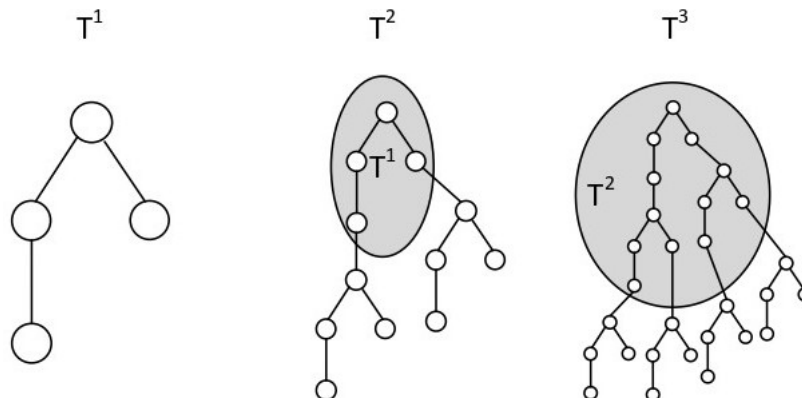


Рис. 2. Пример возведения дерева в степени 1, 2 и 3.

Путь в дереве называется последовательность вершин, в которой две соседние вершины соединены ребром. Все вершины в пути должны быть различны.

Для того, чтобы оценить красоту фейерверка, необходимо определить, какое максимальное количество вершин может содержать путь в дереве, которым представляется фейерверк. На рис. 3 приведен путь в дереве T^2 , содержащий максимальное количество вершин. Таким образом, красота фейерверка с базовым деревом T и мощностью 2 равна 10.

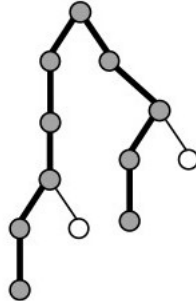


Рис. 3. Путь в дереве T^2 , содержащий максимальное количество вершин.

Требуется написать программу, которая по описанию дерева T и натуральному числу m определяет красоту фейерверка с базовым деревом T и мощностью m .

Формат входных данных

Первая строка входных данных содержит два натуральных числа n и m — количество вершин в базовом дереве фейерверка T и его мощность ($3 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$).

Вторая строка описывает дерево T и содержит $(n - 1)$ целых чисел: $p_2, p_3, \dots, p_n$ — номера родителей вершин $2, 3, \dots, n$ соответственно ($1 \leq p_i \leq i - 1$).

Формат выходных данных

Требуется вывести одно целое число — красоту фейерверка, представляемого деревом T^m .

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи
		n	m	
1	19	$3 \leq n \leq 5000$	$m = 1$	
2	10	$3 \leq n \leq 200\,000$	$m = 1$	1
3	20	$3 \leq n \leq 5000$	$1 \leq m \leq 5000$	1
4	19	$3 \leq n \leq 5000$	$1 \leq m \leq 200\,000$	1, 3
5	32	$3 \leq n \leq 200\,000$	$1 \leq m \leq 200\,000$	1 – 4

Пример

стандартный ввод	стандартный вывод
4 2 1 1 2	10

Задача В. Антенна

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

Для связи с Землёй членам экспедиции на Марс необходимо собрать антенну. Антенна в разобранном состоянии представляет собой n фрагментов, i -й фрагмент представляет собой штангу длиной s_i сантиметров, на которой закреплены m_i перекладин. Каждый фрагмент содержит хотя бы одну перекладину.

У каждой штанги есть начало, в котором расположен штекер, и конец, в котором расположено гнездо. Любые две штанги можно последовательно соединить, присоединив начало одной к концу другой. Для каждой перекладины известно расстояние от начала её штанги в сантиметрах. Для i -го фрагмента это расстояние может быть от 0 до s_i , значение 0 означает, что перекладина находится непосредственно в начале штанги, значение s_i — что она находится непосредственно в конце штанги. Толщиной перекладин и размерами штекера и гнезда следует пренебречь.

Чтобы корректно собрать антенну, необходимо соединить в некотором порядке все n фрагментов, при этом расстояние между любыми двумя соседними перекладинами должно быть одинаковым.

К сожалению, члены экспедиции забыли инструкцию по сборке антенны на Земле, а передать её на Марс не представляется возможным — ведь антенна ещё не собрана. Помогите исследователям!

Требуется определить, в каком порядке необходимо соединить фрагменты антенны, чтобы установить связь с Землей.

Формат входных данных

В первой строке дано одно число n — количество фрагментов ($1 \leq n \leq 100\,000$).

Далее дано описание n фрагментов. В первой строке описания фрагмента даны два целых числа m_i и s_i — количество перекладин и длина штанги в i -м фрагменте ($1 \leq m_i \leq 100\,000$, $0 \leq s_i \leq 10^9$). В следующей строке даны m_i целых чисел $p_{i,j}$ — позиции перекладин, $p_{i,j}$ равно расстоянию в сантиметрах от начала штанги до j -й перекладины на ней ($0 \leq p_{i,1} < p_{i,2} < \dots < p_{i,m_i} \leq s_i$).

Сумма всех m_i не превышает 100 000.

Формат выходных данных

Если собрать антенну указанным образом возможно, в первой строке выведите «Yes», а во второй строке выведите перестановку чисел от 1 до n — номера фрагментов в порядке, в котором их следует соединить, начало каждого следующего фрагмента в этом порядке присоединяется к концу предыдущего фрагмента. Если существует несколько подходящих ответов, можно вывести любой из них.

Если собрать антенну невозможно, в единственной строке выведите «No».

Система оценки

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

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	8	$n \leq 8, m_i = 1, s_i \leq 100$		первая ошибка
2	8	$n \leq 8, s_i \leq 100$	1	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 1 7 3 1 8 6 2 8 1 6	Yes 2 1 3
1 1 7 5	Yes 1
1 3 10 2 5 9	No
3 1 5 3 1 3 3 1 6 3	No
4 1 5 0 1 0 0 1 3 3 1 0 0	Yes 3 2 4 1

Задача С. Автоматизация склада

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

Компания занимается автоматизацией склада. На складе хранятся n видов товаров, пронумерованных от 1 до n , каждый вид товара хранится в своём помещении. Товар вида i хранится в помещении с номером i .

Специальный робот обслуживает запросы по получению товаров со склада. Для доступа в помещения склада робот использует специальные электронные карты. Карты у робота хранятся в специальном отсеке, из которого он может вынуть верхнюю карту. Вынутую карту робот может вернуть в отсек на любое место: на верхнюю позицию, между любыми двумя картами или на самую нижнюю позицию.

Чтобы открыть помещение, робот действует следующим образом. Он вынимает карты из отсека для их хранения и возвращает их обратно в отсек, пока на верхней позиции не окажется карта от помещения, которое ему необходимо открыть. После этого, вынув эту карту, робот использует её, чтобы открыть помещение, и затем также возвращает в отсек для хранения карт. Если суммарно роботу потребовалось вынуть из отсека x карт, включая ту, которой он в итоге открыл помещение, будем говорить, что для открытия помещения робот совершил x действий.

В начале рабочего дня роботу поступил заказ на выдачу m товаров: $a_1, a_2, \dots, a_m$. Робот должен выдать товары именно в этом порядке. Для этого он последовательно выполняет следующие действия: открывает помещение, в котором лежит очередной товар, берет товар, закрывает помещение и выдаёт товар клиенту. После этого робот переходит к выдаче следующего товара.

Исходно электронные карты лежат в отсеке в следующем порядке, от верхней к нижней: $b_1, b_2, \dots, b_n$. Для каждого помещения в отсеке лежит ровно одна карта.

Время выдачи товаров со склада зависит от того, сколько раз суммарно роботу придётся вынимать верхнюю карту из отсека для их хранения, чтобы найти карту от очередного помещения. Необходимо таким образом выбрать места, куда робот должен возвращать вынутые карты, чтобы минимизировать суммарное количество действий робота для открытия помещений.

Требуется написать программу, которая по заданным целым числам n и m , последовательности выдаваемых товаров $a_1, a_2, \dots, a_m$ и начальному положению карт в отсеке для хранения $b_1, b_2, \dots, b_n$ определяет, какое минимальное количество действий придётся совершить роботу, чтобы открыть все помещения в необходимом порядке. Для каждой вынутой карты необходимо также указать позицию, на которую её необходимо вернуть, чтобы добиться оптимального количества действий.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$) — количество видов товаров и количество товаров, которые необходимо выдать со склада.

Вторая строка содержит m целых чисел $a_1, a_2, \dots, a_m$ ($1 \leq a_i \leq n$) — типы товаров, которые необходимо выдать со склада, перечисленные в том порядке, в котором это необходимо сделать.

Третья строка содержит n различных целых чисел $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq n$) — порядок, в котором карты исходно находятся в отсеке для их хранения, перечисленные от верхней к нижней.

Формат выходных данных

Первая строка должна содержать число k — минимальное количество действий, которое потребуется совершить роботу, чтобы выдать товары в заданном порядке.

Далее выведите k чисел. Для каждого действия робота выведите одно число: позицию, на которую ему следует вернуть вынутую карту в отсек для хранения. Если карта возвращается на самую верхнюю позицию, следует вывести 1, если после одной карты, 2, и так далее, для последней позиции следует вывести n .

Если существует несколько способов минимизировать суммарное число действий, выведите любой из них.

Система оценки

Подзадача	Баллы	Доп. ограничения	
1	5	$1 \leq n, m \leq 5 \cdot 10^4$, $n = m$, для всех i верно, что $a_i = b_i$	
2	10	$1 \leq n, m \leq 5 \cdot 10^4$, $n = m$, для всех i верно, что $a_i = b_{n-i+1}$	

Примеры

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

Задача D. Разбиение на пары

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

Космические археологи обнаружили на планете в соседней звездной системе n древних артефактов, которые они пронумеровали от 1 до n . Каждый артефакт имеет k различных параметров, каждый параметр характеризуется целым числом. Артефакт i имеет параметры $a_{i,1}, a_{i,2}, \dots, a_{i,k}$. Оказалось, что первые параметры у всех артефактов различны: для всех $i \neq j$ выполнено $a_{i,1} \neq a_{j,1}$, при этом другие параметры у артефактов могут совпадать.

Учёные также обнаружили текст, в соответствии с которым для активации артефактов их необходимо особым образом разбить на пары и совместить. Разбиение артефактов на пары является корректным, если для каждого t от 1 до k можно выбрать такое число b_t , что оно лежит на отрезке между значениями t -го параметра артефактов каждой пары. То есть, если артефакты i и j образуют пару, должно выполняться условие $a_{i,t} \leq b_t \leq a_{j,t}$ или условие $a_{i,t} \geq b_t \geq a_{j,t}$.

Теперь ученые хотят выяснить, верно ли расшифрован текст. Для этого необходимо проверить, существует ли корректное разбиение артефактов на пары. Каждый артефакт должен войти ровно в одну пару в разбиении.

Требуется написать программу, которая по описанию параметров артефактов определяет, можно ли разбить их на пары таким образом, чтобы для каждого параметра существовало значение, лежащее между значениями этого параметра артефактов каждой пары, и в случае положительного ответа выводит такое разбиение.

Формат входных данных

В первой строке заданы целые числа n и k — количество артефактов и количество параметров ($2 \leq n \leq 2 \cdot 10^5$, n чётно, $1 \leq k \leq 7$).

В следующих n строках задано по k целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,k}$ — параметры артефактов ($-10^9 \leq a_{i,j} \leq 10^9$, все значения $a_{i,1}$ различны).

Формат выходных данных

Выведите «NO», если требуемого разбиения на пары не существует.

В противном случае выведите «YES» в первой строке. Далее выведите $n/2$ строк, в каждой строке выведите по два числа — номера артефактов, из которых следует составить пару. Каждый артефакт должен быть выведен ровно один раз. Если существует несколько корректных разбиений артефактов на пары, разрешается вывести любое из них.

Система оценки

Подзадача	Баллы	Доп. ограничения	
1	10	$2 \leq n \leq 10$	
2	7	$2 \leq n \leq 2 \cdot 10^5$, $k = 1$	
3	15	$2 \leq n \leq 2 \cdot 10^5$, для всех t все значения $a_{i,t}$ различны	
2	15	$2 \leq n \leq 2 \cdot 10^5$, $1 \leq k \leq 2$	

Примеры

стандартный ввод	стандартный вывод
6 2 8 6 1 5 6 3 3 1 4 7 7 2	YES 4 1 5 6 2 3
4 3 1 -1 -1 2 1 1 3 -1 1 4 1 -1	NO

Задача Е. Оптические каналы связи

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

Всего во Флатландии n городов, пронумерованных от 1 до n , столица Флатландии имеет номер 1. Компьютерная сеть Флатландии устроена следующим образом: в каждом городе есть один центр подключения, который может быть связан с некоторыми другими центрами с помощью проводных каналов связи. При этом между любыми двумя городами есть ровно один маршрут по каналам связи, иначе говоря, сеть представляет собой дерево. Для города i , где $i > 1$, обозначим первый город на маршруте от города i до столицы как p_i .

Запланирована модернизация сети Флатландии, в результате которой некоторые каналы связи будут заменены на более современные оптические. Оптические каналы могут быть проложены только вместо существующих проводных. Стоимость замены канала, который соединяет город i с городом p_i , равна w_i . Из-за ограничений технологии любой центр подключения может быть непосредственно подключен оптическими каналами не более чем к k другим центрам.

Министерство связи Флатландии хочет составить такой план модернизации каналов, чтобы после его выполнения связность сети по оптическим каналам связи была как можно выше. Поэтому необходимо выбрать для модернизации как можно больше каналов. Но при этом стоимость модернизации желательно минимизировать, поэтому при равном количестве необходимо выбрать для модернизации каналы с минимальной суммарной стоимостью.

Помогите специалистам министерства выбрать каналы для модернизации.

Формат входных данных

На первой строке ввода находятся два целых числа n и k ($2 \leq n \leq 10^5$, $1 \leq k \leq 100$).

На следующих $n - 1$ строках заданы описания каналов, $(i - 1)$ -я из этих строк содержит два целых числа: p_i и w_i ($1 \leq p_i < i$, $0 \leq w_i \leq 10^9$).

Формат выходных данных

Выведите два целых числа cnt и $cost$: максимальное число каналов, которое удастся модернизировать и минимальную стоимость, за которую можно модернизировать такое число каналов.

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
1	5	$n \leq 15, k = 1, w_i = 0$		первая ошибка
2	5	$n \leq 15, w_i = 0$	1	первая ошибка
3	3	$n \leq 15$	1, 2	первая ошибка
4	7	$k = 1, w_i = 0$	1	первая ошибка
5	5	$k = 1$	1, 4	первая ошибка
6	7	$k \leq 2, w_i = 0$	1, 4	первая ошибка
7	4	$k \leq 2$	1, 4, 5, 6	первая ошибка
8	11	$n \leq 100, w_i = 0$	1, 2	первая ошибка
9	4	$n \leq 100$	1, 2, 3, 8	первая ошибка
10	11	$n \leq 2\,000, w_i = 0$	1, 2, 8	первая ошибка
11	4	$n \leq 2\,000$	1, 2, 3, 8, 9, 10	первая ошибка
12	20	$w_i = 0$	1, 2, 4, 6, 8, 10	первая ошибка
13	14		1–12	первая ошибка

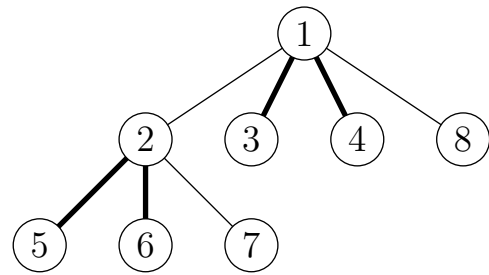
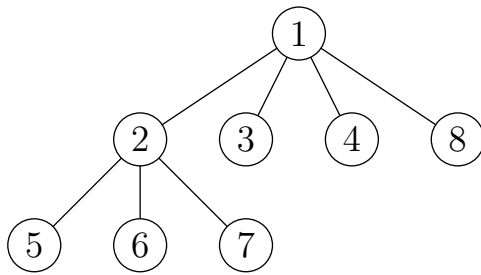
Примеры

стандартный ввод	стандартный вывод
8 2 1 0 1 0 1 0 2 0 2 0 2 0 2 0 1 0	4 0
8 3 1 5 1 2 1 4 2 6 2 7 2 2 2 2 1 6	6 27

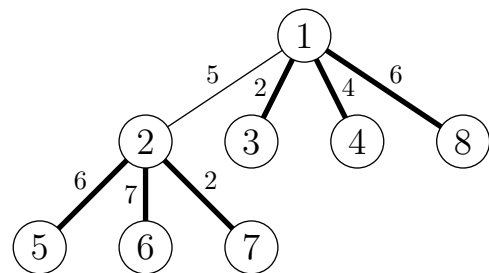
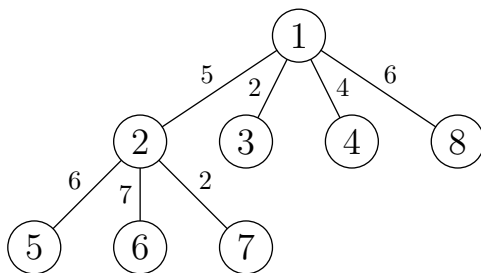
Замечание

Конфигурация сети в первом примере до и после модернизации показана на рисунке ниже. Каналы, которые необходимо модернизировать, показаны жирными линиями. Максимальное число каналов, которое можно модернизировать, равно 4. Стоимость модернизации любого канала равна 0 и не показана.

Есть и другие подходящие решения, в которых модернизируется 4 канала.



Конфигурация сети во втором примере до и после модернизации показана на рисунке ниже. Каналы, которые необходимо модернизировать, показаны жирными линиями. Максимальное число каналов, которое можно модернизировать, равно 6. Стоимость модернизации канала показана рядом с каналом, суммарная стоимость модернизации каналов в оптимальном решении равна 27.



Задача F. Экспедиция

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

Планируется отправить экспедицию к соседней звёздной системе. Были отобраны n кандидатов, пронумерованных от 1 до n , среди которых необходимо выбрать участников экспедиции. Организаторы хотят отправить в экспедицию как можно больше кандидатов.

Среди кандидатов был проведён опрос, в процессе которого каждый мог указать не более, чем одного из остальных кандидатов, с которым он не готов отправиться в экспедицию. Результатом опроса для i -го кандидата является целое число a_i , которое равно номеру кандидата, с которым i -й кандидат не готов отправиться в экспедицию, либо -1 , если i -й кандидат готов отправиться в экспедицию в любом составе.

Теперь организаторы должны выбрать, кто из кандидатов отправится в экспедицию. Решено было выбрать участников экспедиции так, что если туда входит некоторый кандидат i , и $a_i \neq -1$, то туда не входит кандидат a_i . Организаторы хотят выбрать максимальное количество участников экспедиции.

Требуется написать программу, которая по заданным результатам опроса кандидатов определяет максимальное количество кандидатов, которых можно отправить в экспедицию.

Формат входных данных

В первой строке входных данных находится целое число n — количество кандидатов ($1 \leq n \leq 300\,000$).

В следующих n строках даны результаты опроса, i -я из этих строк содержит результат опроса i -го кандидата, целое число a_i ($a_i = -1$ или $1 \leq a_i \leq n$, $a_i \neq i$).

Формат выходных данных

В единственной строке выведите одно целое число — максимальное количество кандидатов, которых можно отправить в экспедицию.

Система оценки

Подзадача	Баллы	Доп. ограничения	
1	19	$n \leq 20$	
2	10	$a_1 = -1$, для $i > 1$ выполнено $a_i = i - 1$	
3	15	для всех i выполнено $a_i < i$	

Примеры

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

Задача G. Хорошие раскраски

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

Назовем раскраску клеток таблицы $n \times m$ *хорошей*, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет.

Иначе говоря, для раскраски не должно быть четверки целых чисел x_1, x_2, y_1, y_2 , что $1 \leq x_1 < x_2 \leq n$, $1 \leq y_1 < y_2 \leq m$, и клетки (x_1, y_1) , (x_2, y_1) , (x_1, y_2) и (x_2, y_2) покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам n , m и c находит любую хорошую раскраску таблицы $n \times m$ в c цветов.

Формат входных данных

В первой строке записаны три целых числа n, m, c ($2 \leq n, m \leq 10$, $2 \leq c \leq 3$).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

Формат выходных данных

Выведите n строк по m чисел в каждой.

В качестве j -го числа i -й строки выведите $a_{i,j}$ — цвет клетки (i, j) ($1 \leq a_{i,j} \leq c$).

Если есть несколько хороших раскрасок, можно вывести любую из них.

Система оценки

Кроме теста из примера в этой задаче 20 тестов, каждый независимо оценивается в 5 баллов. Среди этих тестов в пяти тестах $c = 2$ и в пятнадцати тестах $c = 3$.

Для каждого теста сообщается результат проверки на этом тесте.

Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

Задача Н. Разноцветные точки

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

Рассмотрим n точек на плоскости, пронумерованных от 1 до n , обозначим их как $P_1, P_2, \dots, P_n$, координаты i -й точки (x_i, y_i) .

Рассмотрим следующий процесс. Выберем номер *начальной* точки i и номер *следующей* за ней точки j ($i \neq j$), а также целое число t . После этого номер *прицельной* точки k вычисляется по следующему алгоритму. Рассмотрим вектор $\overrightarrow{P_i P_j}$, направленный из точки P_i в точку P_j . Упорядочим все точки, кроме j -й, по углу, отсчитывая против часовой стрелки от направления вектора, равного $\overrightarrow{P_i P_j}$, отложенного из точки j . При равенстве угла будем упорядочивать точки по возрастанию расстояния до точки j . В качестве точки k выбирается точка, являющаяся t -й в данном порядке при нумерации с единицы. Далее точка j становится начальной, а точка k — следующей за ней, после чего, пользуясь тем же алгоритмом, вычисляется номер прицельной точки. Этот процесс повторяется до бесконечности.

Для лучшего понимания процесса рассмотрим следующий пример. Пусть имеются 6 точек, изображенных на рисунке 1, а $t = 4$. Пусть номер начальной точки равен 1, а номер следующей за ней точки равен 2. Отложим вектор $\overrightarrow{P_1 P_2}$ от точки P_2 и отсортируем все точки, кроме точки P_2 , по углу, отсчитывая против часовой стрелки от направления данного вектора. На рисунке 2 отложенный вектор обозначен пунктирной линией, а также для удобства проведены векторы из точки P_2 во все остальные точки.

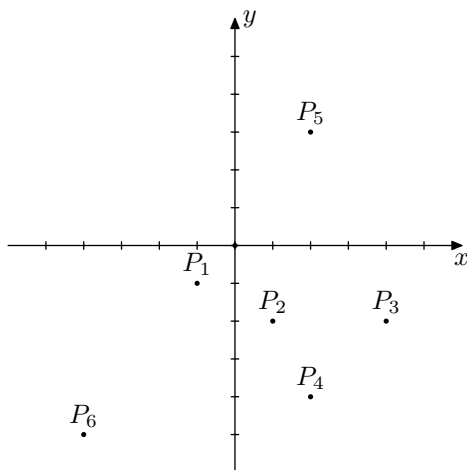


Рисунок 1: Пример множества из 6 точек

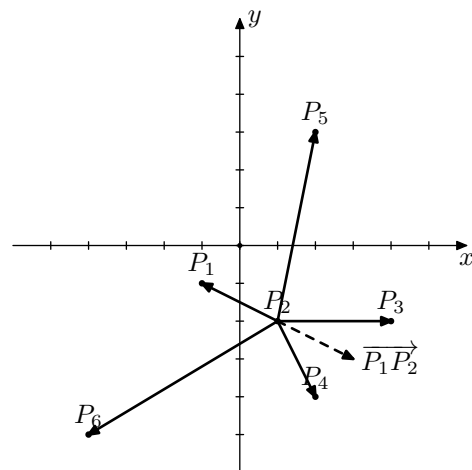


Рисунок 2: Вектор $\overrightarrow{P_1 P_2}$, а также векторы из точки P_2 во все остальные точки

Точки будут упорядочены следующим образом: P_3, P_5, P_1, P_6, P_4 . Таким образом, номер прицельной точки равен 6. Далее точка 2 становится начальной, а точка 6 — следующей.

На рисунке 3 изображен процесс для начальной точки 2 и следующей точки 6. Точки будут упорядочены следующим образом: P_4, P_3, P_2, P_1, P_5 . Обратите внимание, что точка P_1 в этом списке находится раньше, чем точка P_5 , так как расстояние от точки P_1 до точки P_6 меньше, чем расстояние от точки P_5 до точки P_6 . Прицельная точка будет иметь номер 1.

На рисунке 4 изображен процесс для начальной точки 6 и следующей точки 1. Обратите внимание, что в данном случае вектор $\overrightarrow{P_6 P_1}$, отложенный из точки P_1 совпадает с вектором $\overrightarrow{P_1 P_5}$, отложенным из точки P_1 . Эти векторы изображены сплошной линией. Точки будут упорядочены следующим образом: P_5, P_6, P_4, P_2, P_3 . Прицельная точка будет иметь номер 2. Таким образом, далее процесс начнется для начальной точки 1 и следующей точки 2 и заиклится.

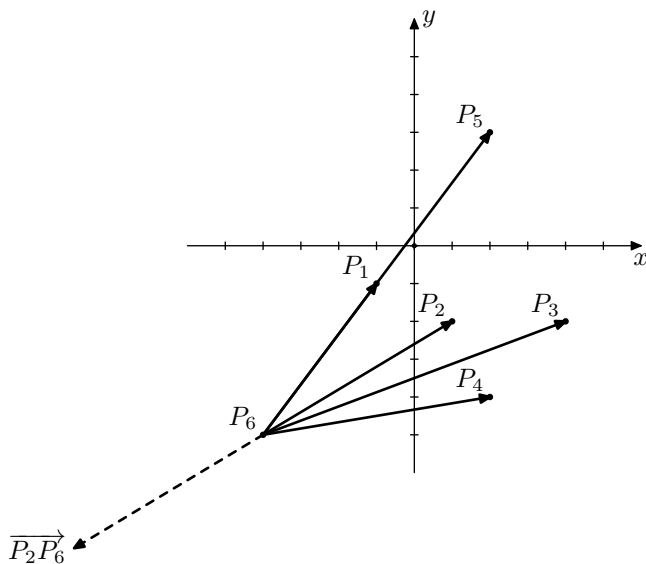


Рисунок 3: Процесс для начальной точки 2 и следующей точки 6

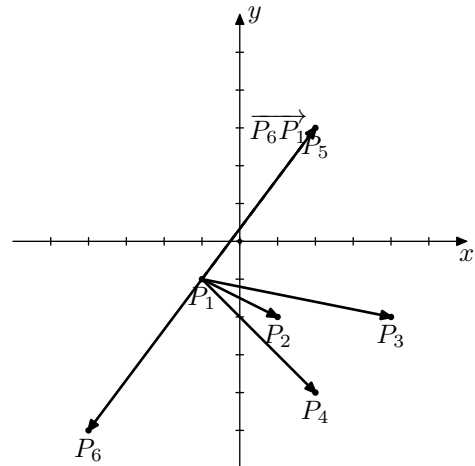


Рисунок 4: Процесс для начальной точки 6 и следующей точки 1

Покрасим каждую из n точек в один из трех цветов. Цвет i -й точки определяется следующим образом:

- Пусть существует такая точка j , что, выбрав точку i в качестве начальной, а точку j в качестве следующей, в результате описанного процесса точка i побывает начальной бесконечное количество раз. В этом случае точка i будет покрашена в **зеленый** цвет.
- Пусть точка i не была покрашена в зеленый цвет и существует такая точка j , что, выбрав точку i в качестве начальной, а точку j в качестве следующей, в результате описанного процесса точка i побывает начальной еще хотя бы один раз. В этом случае точка i будет покрашена в **синий** цвет.
- Пусть точка i не была покрашена ни в зеленый, ни в синий цвет. В этом случае точка i будет покрашена в **красный** цвет.

Для каждой точки определите, в какой цвет ее нужно покрасить.

Формат входных данных

Первая строка содержит два целых числа n и t ($2 \leq n \leq 1\,000$, $1 \leq t \leq n - 1$).

Каждая из следующих n строк содержит два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$). Гарантируется, что никакие две точки не совпадают.

Формат выходных данных

Выведите строку, состоящую из n символов: i -й символ строки должен обозначать цвет i -й точки. Для зеленой точки выведите букву «G», для синей точки — букву «B», а для красной точки — букву «R».

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
7	5	$n \geq 3$, все точки являются вершинами строго выпуклого многоугольника и даны в порядке обхода против часовой стрелки		первая ошибка

Примеры

стандартный ввод	стандартный вывод
6 4 -1 -1 1 -2 4 -2 2 -4 2 3 -4 -5	GGBBRG
2 1 1 1 2 2	GG

Пояснение к примеру

Рассмотрим некоторые точки из первого примера.

Точка P_1 окрашена в зеленый цвет, потому что можно выбрать точку P_2 в качестве следующей, и процесс посетит точку P_1 бесконечное количество раз. Данный пример был рассмотрен выше в условии задачи.

Можно показать, что точка P_3 не является зеленой, однако она является синей, так как можно выбрать точку 1 в качестве следующей, точка 3 окажется начальной еще хотя бы один раз. Процесс для начальной точки 1 и следующей точки 3 проиллюстрирован на рисунках 5, 6 и 7 ниже.

Для начальной точки 3 и следующей точки 1 точки будут упорядочены следующим образом: P_6, P_4, P_2, P_3, P_5 . Точка с номером 3 становится прицельной. Далее для начальной точки 1 и следующей точки 3 точки будут упорядочены следующим образом: P_5, P_1, P_2, P_6, P_4 . Точка с номером 6 становится прицельной. Наконец, для начальной точки 3 и следующей точки 6 точки будут упорядочены следующим образом: P_4, P_3, P_2, P_1, P_5 . Точка с номером 1 становится прицельной. Далее процесс продолжится с начальной точкой 6 и следующей точкой 1. Из примера, описанного выше в условии задачи, мы знаем, что процесс заикнется, посещая точки с номерами 6, 1 и 2.

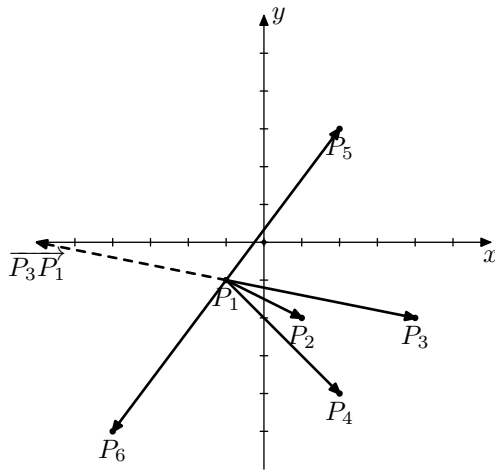


Рисунок 5: Процесс для начальной точки 3 и следующей точки 1

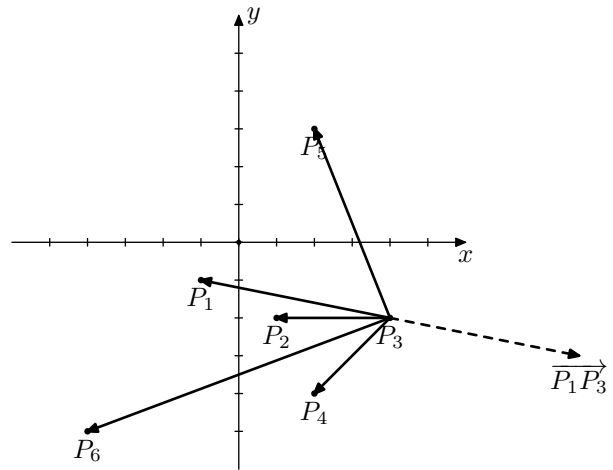


Рисунок 6: Процесс для начальной точки 1 и следующей точки 3

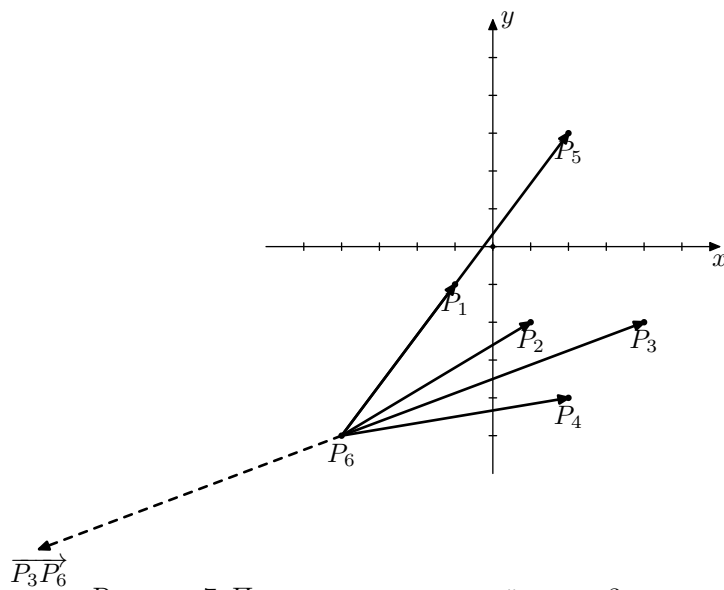


Рисунок 7: Процесс для начальной точки 3 и следующей точки 6

Во втором примере из условия легко показать, что если одна из точек является начальной, а другая — следующей, то прицельной станет точка, которая являлась начальной. Поэтому обе точки будут окрашены в зеленый цвет.