

## Задача 1. Два измерения

Ученые планируют провести важный эксперимент с использованием исследовательского модуля на планете X-2019. В процессе эксперимента будет проведено два измерения: основное и контрольное. Каждое измерение занимает ровно один час и должно начинаться спустя целое число часов после начала работы исследовательского модуля.

Данные эксперимента планируется немедленно передать на орбитальную станцию. Канал связи с орбитальной станцией будет установлен с  $l$ -го по  $r$ -й час от начала работы исследовательского модуля, включительно. Кроме того, согласно плану эксперимента между измерениями планета должна совершить целое число оборотов вокруг своей оси. Планета X-2019 осуществляет оборот вокруг своей оси за  $a$  часов.

Таким образом, если измерения осуществляются на  $i$ -м и  $j$ -м часу, то должно выполняться неравенство  $l \leq i < j \leq r$ , а величина  $(j - i)$  должна быть кратна  $a$ . Теперь учёным необходимо понять, сколько существует различных способов провести измерения.

Требуется написать программу, которая по заданным границам времени измерений  $l$  и  $r$  и периоду обращения планеты вокруг своей оси  $a$  определяет количество возможных способов провести измерения: количество пар целых чисел  $i$  и  $j$ , таких что  $l \leq i < j \leq r$ , и величина  $(j - i)$  кратна  $a$ .

### Формат входных данных

Входные данные содержат три целых числа, по одному на строке:  $l$ ,  $r$  и  $a$  ( $1 \leq l < r \leq 10^9$ ,  $1 \leq a \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число: количество способов провести измерения.

### Примеры

стандартный ввод	стандартный вывод
1 5 2	4
4 9 6	0

### Пояснения к примерам

В первом примере можно провести измерения в следующие пары часов: (1, 3), (1, 5), (2, 4), (3, 5).

Во втором примере продолжительности работы канала связи недостаточно, чтобы провести два измерения.

### Система оценивания

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	30	$1 \leq l < r \leq 100$ , $1 \leq a \leq 100$		полная
2	30	$1 \leq l < r \leq 10^5$ , $1 \leq a \leq 10^5$	1	полная
3	40	$1 \leq l < r \leq 10^9$ , $1 \leq a \leq 10^9$	1, 2	полная

## Задача 2. Полные квадраты

С целью поиска закономерностей иногда полезно сгенерировать длинную последовательность по определенным правилам. Известно, например, что последовательность  $0, 0 + 1, 0 + 1 + 3, 0 + 1 + 3 + 5, \dots, 0 + 1 + 3 + \dots + (2n - 1), \dots$ , составленная из сумм нескольких первых нечетных натуральных чисел, состоит из квадратов целых чисел:  $0, 1, 4, 9, \dots, n^2, \dots$

Обобщим эту последовательность следующим образом: будем использовать вместо начального значения не ноль, а число  $k$ . Получим последовательность:  $k, k + 1, k + 1 + 3, k + 1 + 3 + 5, \dots, k + 1 + 3 + \dots + (2n - 1), \dots$ . В отличие от случая  $k = 0$ , в этой последовательности могут встречаться не только полные квадраты. Необходимо найти минимальное целое неотрицательное число, квадрат которого встречается в этой последовательности.

Требуется написать программу, которая по заданному целому числу  $k$  определяет, квадрат какого минимального неотрицательного целого числа встречается в описанной последовательности, либо выясняет, что в ней вообще не встречается полных квадратов.

### Формат входных данных

В единственной строке содержится целое число  $k$  — начальное число в последовательности ( $-10^{12} \leq k \leq 10^{12}$ ).

Обратите внимание, что для считывания и хранения такого большого числа необходимо использовать 64-битный тип данных.

### Формат выходных данных

Выведите минимальное неотрицательное целое число, квадрат которого встречается в описанной последовательности. Если в последовательности не встречается квадратов целых чисел, выведите «none».

### Примеры

стандартный ввод	стандартный вывод
0	0
-5	2
2	none

### Пояснения к примерам

В первом примере каждое число последовательности является полным квадратом. Минимальный из них — 0,  $0^2 = 0$ .

Во втором примере последовательность начинается так:  $-5, -4, -1, 4, 11, 20, \dots$ . Минимальное неотрицательное целое число, квадрат которого встречается в последовательности — 2,  $2^2 = 4$ .

В третьем примере последовательность начинается так:  $2, 3, 6, 11, 18, \dots$ . В ней нет квадратов целых чисел.

### Система оценивания

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	7	$0 \leq k \leq 1000$		полная
2	10	$0 \leq k \leq 10^5$	1	первая ошибка
3	27	$0 \leq k \leq 10^{12}$	1, 2	первая ошибка
4	7	$-1000 \leq k \leq 1000$	1	полная
5	10	$-10^5 \leq k \leq 10^5$	1, 2, 4	первая ошибка
6	39	$-10^{12} \leq k \leq 10^{12}$	1-5	первая ошибка

### Задача 3. Автоматизация склада

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 требуется 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.

## 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

## Пояснения к примерам

Во втором примере карты в отсеке работа перемещаются следующим образом:

Действие	Перед действием	Извлеченная карта	Открытое помещение	Позиция, куда помещается карта	После действия
1	4, 3, 2, 1	4	4	4	3, 2, 1, 4
2	3, 2, 1, 4	3	–	4	2, 1, 4, 3
3	2, 1, 4, 3	2	–	2	1, 2, 4, 3
4	1, 2, 4, 3	1	1	4	2, 4, 3, 1
5	2, 4, 3, 1	2	2	4	4, 3, 1, 2
6	4, 3, 1, 2	4	4	1	4, 3, 1, 2
7	4, 3, 1, 2	4	4	4	3, 1, 2, 4

## Система оценивания

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

Подз.	Баллы	Ограничения	Необх. подзадачи	Информация о проверке
1	5	$1 \leq n, m \leq 5 \cdot 10^4, n = m,$ для всех $i$ верно, что $a_i = b_i$		полная
2	10	$1 \leq n, m \leq 5 \cdot 10^4, n = m,$ для всех $i$ верно, что $a_i = b_{n-i+1}$		полная
3	31	$1 \leq n, m \leq 2000$		первая ошибка
4	14	$1 \leq n, m \leq 5 \cdot 10^4,$ все $a_i$ различны	1, 2	первая ошибка
5	14	$1 \leq n \leq 5 \cdot 10^4, 1 \leq m \leq 10^5$	1, 2, 3, 4	первая ошибка
6	26	$1 \leq n \leq 3 \cdot 10^5, 1 \leq m \leq 3 \cdot 10^5$	1, 2, 3, 4, 5	первая ошибка

## Задача 4. Машинное обучение

В лаборатории искусственного интеллекта разработали новый метод машинного обучения. В процессе обучения программы используется  $n$  итераций. Каждая итерация заключается в том, что обучаемая программа запускается на некотором обучающем наборе.

Были подготовлены обучающие наборы сложности от 0 до  $k$ . План обучения задаётся массивом целых чисел  $[a_1, a_2, \dots, a_n]$ , где  $a_i$  задаёт сложность набора, используемого на  $i$ -й итерации обучения. Для всех  $i$  от 1 до  $n$  должно выполняться неравенство  $0 \leq a_i \leq k$ .

Выяснилось, что эффективность плана обучения зависит от битовых представлений сложностей обучающих наборов. Для того, чтобы план был эффективным, необходимо, чтобы для любых двух значений  $i$  и  $j$ , где  $1 \leq i < j \leq n$ , выполнялось  $(a_i \text{ and } a_j) = a_i$ . Напомним, что побитовое «и» (and) двух целых неотрицательных чисел устроено следующим образом: запишем оба числа в двоичной системе счисления,  $i$ -й двоичный разряд результата равен 1, если у обоих аргументов он равен 1. Например,  $(14 \text{ and } 7) = (1110_2 \text{ and } 0111_2) = 110_2 = 6$ . Эта операция реализована во всех современных языках программирования, в языках C++, Java и Python она записывается как «&», в Паскале как «and».

Однако постоянное использование наборов одной сложности не даёт прогресса в обучении. Чтобы этого избежать, для плана обучения должны быть выполнены  $m$  требований следующего вида. Каждое требование задаётся двумя числами  $l_i$  и  $r_i$  и означает, что  $a_{l_i} \neq a_{r_i}$ .

Сотрудники лаборатории хотят найти количество эффективных планов, которые удовлетворяют всем требованиям. Так как это число может быть очень большим, нужно найти его остаток от деления на  $10^9 + 7$ .

Требуется написать программу, которая по заданным целым числам  $n$  и  $k$ , а также  $m$  требованиям вида  $l_i, r_i$  определяет количество эффективных планов, которые удовлетворяют всем требованиям, и выводит остаток от деления этого количества на число  $10^9 + 7$ .

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $k$  — количество итераций обучения, количество требований и максимальную сложность обучающего набора ( $1 \leq n \leq 3 \cdot 10^5$ ,  $0 \leq m \leq 3 \cdot 10^5$ ,  $0 \leq k \leq 10^{18}$ ).

Следующие  $m$  строк описывают требования,  $i$ -я строка содержит два целых числа  $l_i, r_i$ , которые означают, что  $a_{l_i} \neq a_{r_i}$  ( $1 \leq l_i < r_i \leq n$ ). Гарантируется, что все требования различны.

### Формат выходных данных

Выведите одно целое число — остаток от деления количества эффективных планов, удовлетворяющих всем требованиям, на число  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
2 0 3	9
3 1 2 1 2	2

### Пояснения к примерам

Все возможные планы для первого теста:  $[0, 0]$ ,  $[0, 1]$ ,  $[0, 2]$ ,  $[0, 3]$ ,  $[1, 1]$ ,  $[1, 3]$ ,  $[2, 2]$ ,  $[2, 3]$ ,  $[3, 3]$ .  
Для второго теста:  $[0, 1, 1]$ ,  $[0, 2, 2]$ .

## Система оценивания

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

Подз.	Баллы	Ограничения	Необх. подзадачи	Информация о проверке
1	8	$1 \leq n \leq 500, m = 0, 0 \leq k \leq 500$		полная
2	20	$1 \leq n \leq 3 \cdot 10^5, m = 0, 0 \leq k \leq 10^7$	1	первая ошибка
3	10	$1 \leq n \leq 3 \cdot 10^5, m = 0, 0 \leq k \leq 10^{18}$	1, 2	первая ошибка
4	8	$1 \leq n \leq 50, 0 \leq m \leq 50, 0 \leq k \leq 50$		первая ошибка
5	16	$1 \leq n \leq 2000, 0 \leq m \leq 2000, 0 \leq k \leq 10^7$	1, 4	первая ошибка
6	6	$1 \leq n \leq 2000, 0 \leq m \leq 2000, 0 \leq k \leq 10^{18}$	1, 4, 5	первая ошибка
7	10	$1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 200, 0 \leq k \leq 10^7$	1, 2, 4	первая ошибка
8	6	$1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 200, 0 \leq k \leq 10^{18}$	1, 2, 3, 4, 7	первая ошибка
9	16	$1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 3 \cdot 10^5, 0 \leq k \leq 10^{18}$	1-8	первая ошибка

## Задача 5. Неисправный марсоход

Марсоход, осуществляющий международную миссию на Марсе, неисправен. Для восстановления его работоспособности необходимо повысить мощность его батареи.

Мощность батареи марсохода задаётся целым положительным числом. Текущая мощность батареи равна  $a$ , для восстановления работоспособности марсохода необходимо повысить её мощность до значения  $b$ . Для изменения мощности батареи на марсоход с Земли можно передавать специальные сигналы двух типов: X и Y. Сигнал типа X увеличивает текущую мощность батареи на 1, а сигнал типа Y увеличивает текущую мощность батареи на 2.

Организаторы миссии хотели бы изменить мощность батареи до необходимой, передав минимальное количество сигналов. К сожалению, из-за особенности устройства марсохода, если мощность батареи оказывается кратна целому числу  $c$ , он окончательно выходит из строя и перестаёт реагировать на сигналы.

Требуется написать программу, которая по заданным начальной мощности батареи  $a$ , необходимой мощности батареи  $b$  и целому числу  $c$  определяет минимальное количество сигналов, которое необходимо передать на марсоход, чтобы восстановить его работоспособность.

### Формат входных данных

Входные данные содержит три целых числа:  $a$ ,  $b$  и  $c$ , по одному на строке ( $1 \leq a < b \leq 10^9$ ,  $2 \leq c \leq 10^9$ ,  $a$  не кратно  $c$ ,  $b$  не кратно  $c$ ).

### Формат выходных данных

Требуется вывести одно целое число: минимальное количество сигналов, которые необходимо передать на марсоход.

### Примеры

стандартный ввод	стандартный вывод
2 7 3	3
4 10 3	4

### Пояснения к примерам

В первом примере можно действовать следующим образом: отправить на марсоход сигналы Y, X, Y. Мощность батареи меняется следующим образом:  $2 \rightarrow 4 \rightarrow 5 \rightarrow 7$ .

Во втором примере можно действовать следующим образом: отправить на марсоход сигналы X, Y, X, Y. Мощность батареи меняется следующим образом:  $4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 10$ .

### Система оценивания

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	25	$1 \leq a < b \leq 15$ , $2 \leq c \leq 15$		полная
2	25	$1 \leq a < b \leq 10^5$ , $2 \leq c \leq 10^5$	1	полная
3	25	$1 \leq a < b \leq 10^9$ , $c = 2$		полная
4	25	$1 \leq a < b \leq 10^9$ , $2 \leq c \leq 10^9$	1, 2, 3	полная

## Задача 6. Интервальные тренировки

В академии физической культуры разработали новый метод интервальных тренировок спортсменов. В соответствии с этим методом спортсмен должен тренироваться каждый день, однако рост нагрузки должен постоянно сменяться её снижением и наоборот.

План тренировки представляет собой набор целых положительных чисел  $a_1, a_2, \dots, a_m$ , где  $a_i$  описывает нагрузку спортсмена в  $i$ -й день. Любые два соседних дня должны иметь различную нагрузку:  $a_i \neq a_{i+1}$ . Чтобы рост нагрузки и её снижение чередовались, для  $i$  от 1 до  $m - 2$  должно выполняться следующее условие: если  $a_i < a_{i+1}$ , то  $a_{i+1} > a_{i+2}$ , если же  $a_i > a_{i+1}$ , то  $a_{i+1} < a_{i+2}$ .

Суммарная нагрузка в процессе выполнения плана должна составлять  $n$ , то есть  $a_1 + a_2 + \dots + a_m = n$ . Ограничения на количество дней в плане нет,  $m$  может быть любым, но нагрузка в первый день тренировок зафиксирована:  $a_1 = k$ .

Прежде чем приступить к тестированию нового метода, руководство академии хочет выяснить, сколько различных планов тренировок удовлетворяет описанным ограничениям.

Требуется написать программу, которая по заданным  $n$  и  $k$  определяет, сколько различных планов тренировок удовлетворяют описанным ограничениям, и выводит остаток от деления количества таких планов на число  $10^9 + 7$ .

### Формат входных данных

В первой строке входных данных находятся целые числа  $n, k$  ( $1 \leq n \leq 5000, 1 \leq k \leq n$ ).

### Формат выходных данных

Выведите одно число: остаток от деления количества планов тренировок на  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
6 2	4
3 3	1

### Пояснения к примерам

В первом примере подходят следующие планы:  $[2, 1, 2, 1]$ ,  $[2, 1, 3]$ ,  $[2, 3, 1]$ ,  $[2, 4]$ .

Во втором примере единственный подходящий план  $[3]$ .

### Система оценивания

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	23	$1 \leq n \leq 10$		полная
2	20	$1 \leq n \leq 30$	1	первая ошибка
3	23	$1 \leq n \leq 500$	1, 2	первая ошибка
4	34	$1 \leq n \leq 5000$	1, 2, 3	только баллы



## Задача 7. Экспедиция

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.

### Примеры

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

### Система оценивания

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

Подз.	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	19	$n \leq 20$		полная
2	10	$a_1 = -1$ , для $i > 1$ выполнено $a_i = i - 1$		первая ошибка
3	15	для всех $i$ выполнено $a_i < i$	2	первая ошибка
4	13	$1 \leq n \leq 2000$	1	первая ошибка
5	43	$1 \leq n \leq 300\,000$	1, 2, 3, 4	первая ошибка

## Задача 8. Разбиение на пары

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.

### Примеры

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

### Пояснения к примерам

В первом примере пары имеют следующие параметры  $(8, 6) - (3, 1)$ ,  $(1, 5) - (7, 2)$ ,  $(6, 3) - (4, 7)$ . При указанном разбиении на пары подойдут, например, значения  $b_1 = 4$ ,  $b_2 = 4$ ,

## Система оценивания

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

Подз.	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	10	$2 \leq n \leq 10$		полная
2	7	$2 \leq n \leq 2 \cdot 10^5, k = 1$		первая ошибка
3	15	$2 \leq n \leq 2 \cdot 10^5$ , для всех $t$ все значения $a_{i,t}$ различны	2	первая ошибка
4	15	$2 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 2$	2	первая ошибка
5	26	$2 \leq n \leq 400, 1 \leq k \leq 7$	1	первая ошибка
6	27	$2 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 7$	1, 2, 3, 4, 5	первая ошибка