

Задача А. Атлеты

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

Вы являетесь организатором крупного турнира, в котором будут участвовать k атлетов. У вас есть n знакомых атлетов, часть из которых вы хотите пригласить на турнир. Про i -го атлета вам известны его сила и выносливость — целые числа p_i и s_i соответственно. Атлет j гарантированно побеждает атлета i только если он строго сильнее и выносливее его, то есть, если $p_j > p_i$ и $s_j > s_i$ одновременно выполнены. В противном случае победить может любой из них.

Турнир проходит следующим образом: пока в нём остаются хотя бы два атлета, из двух произвольных не выбывших атлетов формируется пара и проводится состязание. Победитель состязания остается в турнире, а проигравший выбывает. Так продолжается, пока не будет выявлен победитель турнира, то есть, пока не останется ровно один атлет.

Чтобы турнир был максимально зрелищным и непредсказуемым, вы хотите позвать такой набор из k атлетов, чтобы победитель не был заранее предопределён. Иными словами, если зафиксировать набор из k атлетов и рассмотреть все возможные варианты последовательности состязаний, побед и поражений, то существуют хотя бы два исхода, где победители турнира различаются. Такой набор называется *подходящим*.

Прежде чем определиться с составом турнира, вы хотите выяснить, сколько вообще существует подходящих наборов из ровно k атлетов.

Формат входных данных

Первая строка содержит два целых числа n и k ($1 \leq k \leq n \leq 5\,000$) — количество известных вам атлетов и размер турнира соответственно.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq 10^9$), где i -е число означает силу i -го атлета.

Третья строка содержит n целых чисел s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^9$), где i -е число означает выносливость i -го атлета.

Формат выходных данных

В единственной строке выведите одно число — количество подходящих наборов из ровно k атлетов. Так как ответ может быть довольно большим, выведите его остаток от деления на 998 244 353.

Примеры

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

Задача В. Доказательство неправоты

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

Это интерактивная задача. Не забывайте, что ваша программа должна каждый раз после вывода сбрасывать буфер вывода. Для сброса буфера вывода можно использовать `fflush(stdout)` в C++, `system.out.flush()` в Java, `stdout.flush()` в Python. Если вы используете другой язык программирования, посмотрите в его документации, как выполняется эта операция.

Вася утверждает, что он изобрел алгоритм сортировки n -элементного массива a (в котором **все элементы различны**), который делает меньше, чем $\lceil \log_2(n!) \rceil$ сравнений!

Вася задаст вам не более $\lceil \log_2(n!) \rceil - 1$ запросов вида: сравнить элементы a_i и a_j . Вася утверждает, что данных запросов достаточно, чтобы отсортировать массив a , то есть, чтобы однозначно определить **перестановку позиций** массива a от меньшего элемента к большему: $\text{ord}_1, \text{ord}_2, \dots, \text{ord}_n$.

Ваша задача — найти перестановку позиций, отличающуюся от перестановки Васи, которая также согласуется со всеми запросами Васи.

Протокол взаимодействия

Изначально программа жюри отправит одно целое число n ($3 \leq n \leq 18$) — размер массива.

Далее программа жюри отправит не более $\lceil \log_2(n!) \rceil - 1$ запросов в следующем формате:

- `? i j` — сравнить a_i и a_j ($1 \leq i, j \leq n; i \neq j$).

На каждый запрос жюри ваша программа должна вывести `<`, если $a_i < a_j$, и `>` в противном случае.

После этого программа жюри отправит свой ответ в следующем формате:

- `! ord1 ord2 ... ordn`

В ответ на это ваша программа должна вывести **перестановку**, отличную от перестановки жюри, которая согласуется со всеми предыдущими запросами, в аналогичном формате. Если возможных перестановок несколько, выведите любую из них. Обратите внимание, что **массив a выводить не нужно!**

Если ваш ответ правильный, программа жюри отправит строку `CORRECT` в отдельной строке, указывая, что вы можете завершить программу. Если ответ неверен, программа жюри отправит строку `ERROR` в отдельной строке.

Если ваша программа нарушит формат запросов, жюри ответит строкой `ERROR`. В таком случае ваша программа должна немедленно завершиться, чтобы получить вердикт **Неправильный ответ**. В противном случае вы можете получить произвольный вердикт, потому что ваше решение продолжит читать из закрытого потока.

Не забывайте сбрасывать буфер вывода каждый раз, когда что-то выводите!

Пример

стандартный ввод	стандартный вывод
4	
? 2 1	
	>
? 4 3	
	>
? 4 2	
	<
? 1 4	
	<
! 1 3 4 2	
	! 3 1 4 2
CORRECT	

Замечание

В примере из условия подходящим массивом a мог быть, например, массив 3 9 0 7. Этот массив подходит под все результаты сравнений, однако перестановка его элементов, предложенная алгоритмом Васи, не является его сортировкой.

Обратите внимание, что пробелы в примерах добавлены только для улучшения читаемости. При проверке решений интерактор пустых строк выводить не будет.

Задача С. Кафе для мафии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

В городе NN закончилось строительство кафе для мафии. Столики в этом кафе одноместные и пронумерованы числами от 1 до n в порядке привлекательности, где столик 1 является самым привлекательным, а столик n — самым непривлекательным. Для удобства также пронумеруем членов мафии числами от 1 до n в порядке их значимости, так что номер 1 — это самый значимый босс мафии, а номер n — самый незначительный новобранец.

Мафиози — очень занятые люди, и у i -го из них есть лишь один временной промежуток в расписании, когда он может посетить кафе — с момента времени a_i до момента времени b_i включительно, и он хочет провести в кафе всё это время.

Администратор кафе планирует рассадку людей за столики, стараясь самым значимым членам мафии отдать самые привлекательные столики. Мафиози номер 1 он посадит за стол номер 1, тем самым заняв этот стол в промежуток времени с a_1 до b_1 . Мафиози номер 2 он попытается также посадить за стол номер 1, но если отрезки $[a_1, b_1]$ и $[a_2, b_2]$ пересекаются, то сделать это не получится, и мафиози номер 2 достанется стол номер 2.

Таким образом, мафиози номер i назначают столик сразу после мафиози номер $i - 1$. Из всех свободных (в промежуток времени с a_i до b_i) столиков мафиози номер i посадят за столик с наименьшим номером. Обратите внимание, что обе границы отрезка включены в него, то есть, если для каких-то двух различных мафиози i и j верно, что $b_i = a_j$, то им обязательно должны достаться разные столики.

Помогите администратору составить рассадку.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 250\,000$) — число мафиози и столиков соответственно.

Далее в наборе входных данных следует n строк, i -я из них содержит два целых числа a_i, b_i ($1 \leq a_i \leq b_i \leq 2n$), означающих, что i -й мафиози планирует быть в кафе в период времени с a_i по b_i включительно.

Гарантируется, что сумма n по всем тестовым наборам не превосходят 250 000.

Формат выходных данных

Для каждого набора входных данных выведите n целых чисел c_1, c_2, \dots, c_n , где c_i — это номер столика, за который сядет i -й мафиози.

Пример

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

Задача D. Три подотрезка

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

Дан массив целых чисел a_1, a_2, \dots, a_n , все числа в массиве от 1 до m .

Посчитайте количество способов выбрать три непустых непересекающихся отрезка a так, чтобы количество вхождений каждого числа от 1 до m в объединение этих трёх отрезков было одним и тем же **чётным** числом.

Формат входных данных

Первая строка содержит два целых числа n и m ($3 \leq n \leq 300; 1 \leq m \leq n$) — размер массива и ограничение на максимальный элемент в нём соответственно.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq m$) — элементы массива a .

Формат выходных данных

В единственной строке выведите ответ на задачу — количество способов выбрать три непустых непересекающихся отрезка массива так, чтобы количество вхождений каждого числа от 1 до m в объединение этих трёх отрезков было одним и тем же **чётным** числом.

Примеры

стандартный ввод	стандартный вывод
5 2 2 1 2 2 1	7
12 3 1 2 3 2 1 3 3 1 1 2 3 2	278

Замечание

В первом тесте все хорошие объединения выглядят так:

- [2] [1] 2 [2 1];
- [2] [1 2] 2 [1];
- [2 1] [2] 2 [1];
- [2 1] 2 [2] [1];
- 2 [1] [2] [2 1];
- 2 [1] [2 2] [1];
- 2 [1 2] [2] [1];

Для второго теста приведём лишь некоторые из хороших объединений:

- [1 2 3 2] [1 3 3 1 1 2] [3 2];
- 1 2 [3 2] 1 [3] 3 [1 1 2] 3 2;
- [1 2 3] 2 1 3 [3 1] 1 2 3 [2];
- [1] [2 3 2] [1 3] 3 1 1 2 3 2;
- [1] 2 3 2 1 3 [3 1] 1 [2 3 2];
- 1 [2 3] [2 1 3] 3 1 [1] 2 3 2;

Задача Е. Кратчайший некратчайший путь

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дан связный неориентированный граф, состоящий из n вершин и m ребер. В графе отсутствуют петли и кратные рёбра. Найдите в этом графе любой путь P между заданными вершинами s и t такой, что:

1. P не содержит никакую вершину графа дважды;
2. P содержит строго больше вершин, чем кратчайший путь между s и t ;
3. P содержит наименьшее число вершин среди всех путей, удовлетворяющих условиям 1 и 2.

Формат входных данных

Первая строка содержит четыре целых числа n, m, s, t ($2 \leq n \leq 2000$; $n - 1 \leq m \leq 25\,000$; $1 \leq s, t \leq n, s \neq t$) — число вершин и рёбер в графе и номера концов пути P соответственно.

Следующие m строк описывают ребра: ребро i задано парой вершин a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$), где a_i и b_i — это вершины, которые соединяет i -е ребро.

Гарантируется, что граф связный и не содержит петель и кратных рёбер.

Формат выходных данных

Если искомого пути P не существует, в единственной строке выведите -1 .

Иначе в первой строке выведите число k — количество вершин в пути P . Во второй строке выведите k различных целых чисел, v_1, v_2, \dots, v_k ($1 \leq v_i \leq n, v_1 = s, v_k = t, v_i$ и v_{i+1} соединены ребром для любого $1 \leq i < k$) — номера вершин в пути P в порядке следования.

Если ответов несколько, выведите любой из них.

Примеры

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

Задача F. Почти двудольный граф

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

Дан неориентированный граф, состоящий из n вершин и m рёбер. В графе отсутствуют петли и кратные рёбра. Не гарантируется, что данный граф — связный.

Напомним, что граф называется двудольным, если можно покрасить все вершины в два цвета так, чтобы ни одно ребро не соединяло две вершины одного цвета.

Ваша задача — определить, можно ли удалить не более k рёбер таким образом, чтобы получившийся граф стал двудольным.

Формат входных данных

Первая строка содержит три целых числа n , m и k ($2 \leq n \leq 1\,000$; $1 \leq m \leq 1\,000$; $1 \leq k \leq 10$) — количество вершин и рёбер в графе, и максимальное число рёбер, которое можно удалить.

Далее следуют m строк, i -я из них содержит два целых числа x_i и y_i ($1 \leq x_i, y_i \leq n$; $x_i \neq y_i$), означающих ребро между вершинами x_i и y_i . В заданном графе отсутствуют петли и кратные рёбра.

Формат выходных данных

В первой строке выведите количество рёбер (не более k), которые необходимо удалить, чтобы граф стал двудольным. Во второй строке выведите k различных чисел — номера рёбер для удаления.

Если такого набора рёбер размера не более k не существует, в единственной строке выведите -1 .

Если существует несколько подходящих наборов рёбер, то выведите любой из них. Номера рёбер можно выводить в произвольном порядке. Рёбра пронумерованы целыми числами от 1 до m в порядке их следования во входных данных.

Примеры

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