

Задача А. Выражение

Петя учится в школе и очень любит математику. Уже несколько занятий они с классом проходят арифметические выражения. На последнем уроке учительница написала на доске три положительных целых числа a , b , c . Задание заключалось в том, чтобы расставить между этими числами знаки операций '+' и '*', а также, возможно, скобки. Значение получившегося выражения должно быть как можно больше. Рассмотрим пример: пусть учительница выписала на доску числа 1, 2 и 3. Вот некоторые варианты расстановки знаков и скобок:

- $1+2*3=7$
- $1*(2+3)=5$
- $1*2*3=6$
- $(1+2)*3=9$

Обратите внимание на то, что знаки операций можно вставлять только между a и b , а также между b и c , то есть нельзя менять числа местами. Так, в приведенном примере нельзя получить выражение $(1+3)*2$.

Легко убедиться, что максимальное значение, которое можно получить, — это 9.

Ваша задача — по заданным a , b и c вывести, какое максимальное значение выражения можно получить.

Формат входных данных

Во входных данных записаны три целых числа a , b и c , каждое в отдельной строке ($1 \leq a, b, c \leq 10$).

Формат выходных данных

Выведите максимальное значение выражения, которое можно получить.

Примеры

ввод с клавиатуры	вывод на экран
1 2 3	9
2 10 3	60

Задача В. Разности

Маленькому Васе на день рождения подарили набор из n целых чисел: $1, 2, \dots, n$. На радостях мальчик решил расположить все числа в ряд таким образом, чтобы модуль разности любых двух соседних чисел был не меньше $\lfloor \frac{n}{2} \rfloor$.

Под выражением $\lfloor \frac{n}{2} \rfloor$ понимается целая часть (частное) от деления числа n на 2. Например, $\lfloor \frac{7}{2} \rfloor = 3$, $\lfloor \frac{10}{2} \rfloor = 5$.

После десятка неудачных попыток Вася все же сумел расположить числа нужным образом. А справитесь ли вы?

Формат входных данных

В единственной строке записано целое число n ($1 \leq n \leq 100$) — количество чисел в подаренном наборе.

Формат выходных данных

Выведите удовлетворяющую условию задачи последовательность из n различных целых чисел, лежащих в диапазоне от 1 до n включительно. Если существует несколько подходящих последовательностей, разрешается вывести любую из них. Гарантируется, что хотя бы одна такая последовательность существует.

Примеры

ввод с клавиатуры	вывод на экран
3	2 1 3
4	3 1 4 2

Задача С. Башни

Как известно, все дети в Берляндии любят играть с кубиками. У маленького Пети имеется n башен, состоящих из кубиков одинакового размера. Башня под номером i представляет собой a_i кубиков, поставленных друг на друга. *Неустойчивостью* набора башен Петя называет величину, равную разности высот самой высокой и самой низкой башни. К примеру, если Петя построил из кубиков пять башен с высотами (8, 3, 2, 6, 3), то неустойчивость этого набора равна 6 (самая высокая башня имеет высоту 8, самая низкая — высоту 2).

Мальчик хочет, чтобы неустойчивость его набора башен была как можно меньше. Все, что он может сделать, это несколько раз проделать следующую операцию: взять верхний кубик с какой-то из башен и положить его сверху на какую-то другую башню из своего набора. Обратите внимание, что Петя никогда не будет класть кубик на ту же башню, с которой тот был снят, потому что считает это пустой тратой времени.

Прежде чем отправиться в школу, мальчик успеет произвести не более k таких операций. Петя не хочет опоздать на урок, поэтому вам придется помочь ему выполнить эту задачу.

Формат входных данных

В первой строке через пробел записаны два целых положительных числа n и k ($1 \leq n \leq 100$, $1 \leq k \leq 1000$) — количество башен в имеющемся наборе и максимальное число операций, которые Петя может произвести. Во второй строке через пробел записаны n целых положительных чисел a_i ($1 \leq a_i \leq 10^4$) — исходные высоты башен.

Формат выходных данных

В первой строке выведите через пробел два целых неотрицательных числа s и m ($m \leq k$). Первое из чисел — это величина минимально возможной неустойчивости, которую можно достичь, применив не более k операций, а второе — количество операций, необходимых для этого.

Затем в m строках выведите описание каждой из операций в виде двух целых положительных чисел i и j , каждое из которых лежит в пределах от 1 до n . Они обозначают, что Петя переложил верхний кубик с i -й башни на j -ю ($i \neq j$). Обратите внимание, что в процессе выполнения операций высоты некоторых башен могут стать равны нулю.

Если существует несколько корректных последовательностей операций, при которых достигается минимально возможная неустойчивость, разрешается вывести любую из них.

Примеры

ввод с клавиатуры	вывод на экран
3 2 5 8 5	0 2 2 1 2 3
3 4 2 2 4	1 1 3 2
5 3 8 3 2 6 3	3 3 1 3 1 2 1 3

Замечание

В первом примере осуществляется два перекладывания, со второй башни на третью и со второй на первую. После этого высоты башен становятся одинаковыми и равными 6.

Задача D. Экзамены

Студент Валера учится на первом курсе университета. Скоро у него сессия, и ему предстоит сдать ровно n экзаменов. Валера — умный парень, поэтому он сможет сдать любой экзамен с первого раза. Кроме того, он может сдавать несколько экзаменов в один день и в любом порядке.

Согласно расписанию, экзамен по i -му предмету нужно сдать в день с номером a_i . Однако Валера договорился с каждым преподавателем, и преподаватель i -го предмета разрешил организовать досрочную сдачу своего экзамена в день b_i ($b_i < a_i$). Таким образом, Валера может сдать экзамен по i -му предмету либо в день a_i , либо в день b_i . Все преподаватели ставят запись о сдаче экзамена в зачетную книжку в день фактической сдачи экзамена и датируют эту запись числом a_i .

Валера считает, что будет достаточно странно, если записи в зачетной книжке будут идти не в порядке неубывания даты. Поэтому Валера просит вас помочь ему. Найдите минимально возможный номер дня, когда Валера сможет сдать последний экзамен, если он будет сдавать экзамены так, чтобы все записи в его зачетной книжке шли в порядке неубывания даты.

Формат входных данных

В первой строке записано единственное целое положительное число n ($1 \leq n \leq 5000$) — количество экзаменов, которые будет сдавать Валера.

В каждой из следующих n строк записано по два целых положительных числа через пробел a_i и b_i ($1 \leq b_i < a_i \leq 10^9$) — дата сдачи по расписанию и досрочная дата сдачи i -го экзамена соответственно.

Формат выходных данных

Выведите единственное целое число — минимально возможный номер дня, когда Валера сможет сдать последний экзамен, если он будет сдавать экзамены так, чтобы все записи в его зачетной книжке шли в порядке неубывания даты.

Примеры

ввод с клавиатуры	вывод на экран
3 5 2 3 1 4 2	2
3 6 1 5 2 4 3	6

Замечание

В первом примере Валера сначала сдаст экзамен по второму предмету в первый день (в зачетную книжку заносится дата сдачи по расписанию, то есть 3). На следующий день сначала он сдаст экзамен по третьему предмету (будет добавлена запись, датированная днем 4), а после него в тот же день сдаст экзамен по первому предмету (в зачетную книжку отметка будет проставлена с днем 5). Таким образом, последний экзамен Валера сдаст во второй день, и даты в зачетной книжке будут идти в порядке неубывания: 3, 4, 5.

Во втором примере Валера сначала сдаст экзамен по третьему предмету в четвертый день, после него экзамен по второму предмету в пятый день. После этого в шестой день Валера сдаст экзамен по первому предмету.

Задача E. Две матрицы

Валера учится в университете на первом курсе. На одном из занятий преподаватель нарисовал на доске две матрицы из n строк и m столбцов, состоящих из целых чисел. После этого он попросил проверить утверждение: верно ли, что из первой матрицы можно получить вторую матрицу, используя следующие операции:

- прибавить единицу ко всем элементам произвольной строки первой матрицы;
- вычесть единицу из всех элементов произвольного столбца первой матрицы.

После долгих безуспешных попыток решить эту задачу, Валера обратился за помощью к вам. Помогите ему проверить, возможно ли получить вторую матрицу из первой, используя вышеперечисленные операции, или сообщите, что решения не существует.

Обратите внимание, что если выполнять все операции последовательно, то в какой-то момент в матрице могут получаться отрицательные числа.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leq 100$) — число строк и столбцов в матрицах соответственно.

В каждой из следующих n строк записаны m целых чисел через пробел: в i -й из этих строк на j -м месте записан элемент $a_{i,j}$ ($0 \leq a_{i,j} \leq 10^9$), лежащий на пересечении i -й строки и j -го столбца первой матрицы.

В каждой из следующих n строк записаны m целых чисел через пробел: в i -й из этих строк на j -м месте записан элемент $b_{i,j}$ ($0 \leq b_{i,j} \leq 10^9$), лежащий на пересечении i -й строки и j -го столбца второй матрицы.

Формат выходных данных

Если из первой матрицы получить вторую невозможно, то в единственной строке выведите «NO» (без кавычек).

В случае, если из первой матрицы можно получить вторую с помощью перечисленных действий, выведите в первой строке «YES» (без кавычек). Затем для каждой строки (столбца) выведите, сколько раз к этой строке (столбцу) будет применена соответствующая операция. А именно, во второй строке выведите n целых чисел x_1, x_2, \dots, x_n , где x_i означает, сколько раз следует прибавить единицу к i -й строке. В третьей строке выведите m целых чисел y_1, y_2, \dots, y_m , где y_j означает, сколько раз следует вычесть единицу из j -го столбца. Допускается вывести любое корректное решение, в котором $0 \leq x_i \leq 10^{18}$ и $0 \leq y_j \leq 10^{18}$. Гарантируется, что если ответ существует, то существует и ответ, удовлетворяющий данным ограничениям.

Примеры

ввод с клавиатуры	вывод на экран
2 3 1 1 1 1 1 1 2 3 1 1 2 0	YES 2 1 1 0 2
2 2 1 1 1 1 2 3 3 2	NO

Замечание

Гарантируется, что не менее чем в 50% тестов жюри элементы обеих матриц находятся в диапазоне от 0 до 1000.

Задача F. Телефонная книга

Вы купили себе новый мобильный телефон. Однако, нельзя просто так взять и начать пользоваться новым телефоном, потому что в нем не записано ни одного номера. Поэтому первым делом вы добавили в список контактов номера телефонов всех ваших друзей и близких. Таким образом в вашей телефонной книге оказались n записей. Каждая запись представляет собой пару из имени и номера телефона.

Как и многие современные смартфоны, ваш телефон умеет догадываться по префиксу набираемого номера или имени, кому вы хотите позвонить. Более того, он выдает вам список предлагаемых контактов в порядке релевантности. Это означает, что первыми в этом списке будут располагаться номера телефонов тех людей, которым вы чаще звонили. В случае равенства количества совершенных звонков, контакты сортируются лексикографически по имени. Причем на экране умещается не более k первых контактов из этого упорядоченного списка.

Сегодня первый день, когда вы начали пользоваться вашим новым телефоном. Вы еще не успели никому позвонить, однако вы точно знаете, что собираетесь совершить некоторое количество звонков, а также периодически наблюдать, как работает описанная функциональность вашего телефона. Более формально, вам нужно выполнить m операций двух типов:

1. “call query” — совершить звонок одному из ваших n контактов, где query — это либо номер, либо имя того, кому вы хотите позвонить;
2. “find query” — узнать список из не более k предлагаемых телефоном контактов в порядке релевантности, где query — это префикс либо номера, либо имени некоторого количества ваших контактов (возможно нулевого). Под префиксом номера или имени понимается его некоторый непустой начальный фрагмент. Возможно, префикс совпадает с номером или именем целиком.

Для каждой операции второго типа выведите искомый список. Внимательно ознакомьтесь со входными и выходными данными, чтобы лучше изучить описанные операции.

Формат входных данных

В первой строке через пробел записаны два целых числа n и k ($1 \leq n, k \leq 100$) — количество записей в телефонной книге и размер экрана телефона соответственно. В каждой из следующих n строк задано описание телефонной книги. В i -й строке через пробел задана пара $name_i$ и $number_i$ — имя и номер телефона i -го контакта соответственно. Гарантируется, что все имена являются непустыми строками длины не более 100, состоящими из строчных и прописных букв латинского алфавита. Регистр букв имеет значение: имена, различающиеся только регистром букв (например, «Dmitry» и «dmitry»), считаются различными. Гарантируется, что все номера телефонов являются последовательностями из 11 цифр (лидирующие нули допускаются). Гарантируется, что все имена и номера телефонов различны.

В следующей строке задано единственное целое число m ($1 \leq m \leq 100$) — количество операций. В каждой из следующих m строк задано описание очередной операции. В i -й строке через пробел задана пара $type_i$ и $query_i$ — тип и строка запроса i -й операции соответственно. Если $type_i = \text{“call”}$, то i -я операция — первого типа. Если $type_i = \text{“find”}$, то i -я операция — второго типа. Другие типы операций отсутствуют. Гарантируется, что для всех операций первого типа параметр $query$ является корректным именем, либо номером телефона одного из n контактов. Гарантируется, что для всех операций второго типа параметр $query$ является непустой строкой длины не более 100, состоящей либо только из цифр, либо только из строчных и прописных букв латинского алфавита.

Формат выходных данных

Для каждой операции второго типа в отдельной строке выведите искомый список — упорядоченный по релевантности список предлагаемых контактов длины не более k . Сначала выведите размер, а затем сами элементы списка, разделенные единственным пробелом.

В первую очередь контакты сортируются по невозрастанию количества совершенных звонков, во вторую очередь — лексикографически по имени (как сортировка по алфавиту в словарях). То

есть строка «aba» лексикографически меньше строки «ba», а строка «a» лексикографически меньше строки «aa». Строчные и прописные буквы следует отличать (в частности это означает, что символ 'A' лексикографически меньше символа 'a').

Примеры

ввод с клавиатуры	вывод на экран
6 2	2 Maxim Danil
Dmitry 89198298406	2 Ilya Maxim
Maxim 89271510591	0
Ilya 89053884155	2 Danil Dmitry
Danil 89603593953	2 Dmitry Danil
Pavel 89603596707	2 Danil Dmitry
Vitaly 89093311480	2 Danil Pavel
13	1 Vitaly
call Maxim	1 Pavel
find 89	
call Ilya	
find 8	
find 100500	
find D	
call Dmitry	
find D	
call 89603593953	
find D	
find 8960359	
find Vit	
find 89603596707	

Задача G. Прыжки в длину

Валерий — учитель физической культуры в одной из школ Берляндии. Скоро школьникам сдавать норматив по прыжкам в длину, а Валерий потерял свою любимую линейку!

Однако нет поводов для разочарования, поскольку Валерий нашел другую линейку длины l сантиметров. На линейку уже нанесены n отметок, с помощью которых можно производить измерения. Будем считать, что отметки пронумерованы числами от 1 до n в порядке их следования от начала линейки к ее концу. Первая отметка совпадает с началом линейки и обозначает начало отсчета. Последняя отметка совпадает с концом линейки и находится на расстоянии l от начала отсчета. Такую линейку можно задать возрастающей последовательностью a_1, a_2, \dots, a_n , где a_i означает расстояние i -й отметки от начала отсчета ($a_1 = 0, a_n = l$).

Валерий считает, что с помощью линейки можно отмерить расстояние d сантиметров, если существует пара целых чисел i и j ($1 \leq i \leq j \leq n$), такая, что расстояние между i -й и j -й отметкой в точности равно d (иными словами, $a_j - a_i = d$).

Согласно нормам, девочки должны уметь прыгать в длину на расстояние не менее чем x сантиметров, а мальчики должны уметь прыгать в длину на расстояние не менее чем y ($x < y$) сантиметров. Для проверки способностей детей Валерию нужна линейка, с помощью которой можно отмерить каждое из расстояний x и y .

Ваша задача — определить, какое минимальное количество дополнительных отметок необходимо нанести на линейку так, чтобы с их помощью можно было отмерить расстояния x и y . Валерий может наносить отметки на любом целочисленном неотрицательном расстоянии от начала отсчета, не превышающем длину линейки.

Формат входных данных

В первой строке записано четыре целых положительных числа через пробел n, l, x, y ($2 \leq n \leq 2000, 2 \leq l \leq 10^8, 1 \leq x < y \leq l$) — количество отметок, длина линейки и нормативы для девочек и мальчиков соответственно.

Во второй строке записана последовательность из n целых чисел a_1, a_2, \dots, a_n ($0 = a_1 < a_2 < \dots < a_n = l$), где a_i означает расстояние от i -й отметки до начала отсчета.

Формат выходных данных

В первой строке выведите единственное целое неотрицательное число v — минимальное количество отметок, которые нужно дополнительно нанести на линейку.

Во второй строке выведите через пробел v целых чисел p_1, p_2, \dots, p_v ($0 \leq p_i \leq l$). Число p_i означает, что i -ю отметку следует нанести на расстоянии p_i сантиметров от начала отсчета. Отметки выводите в любом порядке. Если оптимальных решений несколько, выведите любое из них.

Примеры

ввод с клавиатуры	вывод на экран
3 250 185 230 0 185 250	1 230
4 250 185 230 0 20 185 250	0
2 300 185 230 0 300	2 185 230

Замечание

В первом примере изначально нельзя отмерить расстояние 230 сантиметров. Для этого достаточно добавить отметку 20 сантиметров или отметку 230 сантиметров.

Во втором примере с помощью линейки уже можно отмерить расстояния 185 и 230 сантиметров, поэтому добавлять новые отметки не нужно.

В третьем примере на линейку нанесены только отметки начала и конца. Нам потребуется добавить две отметки, чтобы проверка способностей детей стала возможной.

Задача Н. Катаемся на лифте

Представьте себе, что вы находитесь в здании, в котором есть ровно n этажей. Для перемещения между этажами в здании есть лифт.

Пронумеруем этажи снизу вверх целыми числами от 1 до n . Сейчас вы находитесь на этаже с номером a . Вам очень скучно, поэтому вы хотите покататься на лифте. На этаже с номером b находится секретная лаборатория, вход в которую вам запрещен. Однако вы уже настроились и решили последовательно совершить k поездок на лифте.

Предположим, что в данный момент вы находитесь на этаже с номером x (изначально вы находились на этаже a). Для очередной поездки между этажами вы выбираете некоторый этаж с номером y ($y \neq x$), и лифт везет вас на этот этаж. Поскольку вам нельзя посещать этаж b с секретной лабораторией, то вы решили, что расстояние от текущего этажа x до выбранного этажа y должно быть строго меньше, чем расстояние от текущего этажа x до этажа b с секретной лабораторией. Формально, это означает, что должно выполняться неравенство $|x - y| < |x - b|$. После того, как лифт успешно привез вас на этаж y , вы выписываете себе в блокнот число y .

Ваша задача — определить количество различных последовательностей чисел, которые вы могли выписать себе в блокнот в результате k поездок на лифте. Поскольку искомое количество может быть очень большим, найдите остаток от деления этого количества на 1000000007 ($10^9 + 7$).

Формат входных данных

В первой строке входных данных записаны четыре целых положительных числа через пробел n, a, b, k ($2 \leq n \leq 5000, 1 \leq k \leq 5000, 1 \leq a, b \leq n, a \neq b$).

Формат выходных данных

Выведите единственное число — остаток от деления искомого количества последовательностей на 1000000007 ($10^9 + 7$).

Примеры

ввод с клавиатуры	вывод на экран
5 2 4 1	2
5 2 4 2	2
5 3 4 1	0

Замечание

Две последовательности p_1, p_2, \dots, p_k и q_1, q_2, \dots, q_k называются *различными*, если существует такое целое число j ($1 \leq j \leq k$), что $p_j \neq q_j$.

Пояснения к примерам:

1. В первом примере после первой поездки вы можете оказаться либо на этаже 1, либо на этаже 3, поскольку $|1 - 2| < |2 - 4|$ и $|3 - 2| < |2 - 4|$.
2. Во втором примере всего две возможных последовательности: $(1, 2); (1, 3)$. Вы не можете выбрать этаж 3 для первой поездки, потому что в этом случае никакой из этажей не подойдет для второй поездки.
3. В третьем примере искомым последовательностей не существует, поскольку вы не можете выбрать этаж для первой поездки.

Задача I. Разбиение

Недавно Петя задумался над решением следующей задачи. Дан набор положительных целых чисел. Нужно разбить его на два набора таким образом, чтобы суммы двух наборов отличались как можно меньше. К примеру, набор $\{3, 3, 4, 5, 8\}$ следует разбить на $\{3, 3, 5\}$ и $\{4, 8\}$.

Петя потратил не один день на размышления, но так и не смог прийти к решению. Когда он полез за ответом в интернет, он выяснил, что его неудача легко объяснима: эта задача относится к так называемому классу NP-трудных задач. Не вдаваясь в определения, Петя понял: если задача принадлежит этому классу, то эффективного решения (отличного от перебора), вероятно, не существует.

Однако, Петя узнал, что существуют различные эвристики (приближенные алгоритмы), которые дают решение, близкое к оптимальному. Пете приглянулся так называемый разностный метод, который он теперь хочет освоить.

К сожалению, в той статье, которую читал Петя, метод был описан не очень подробно. А именно, был приведен только алгоритм получения приближенной разности между суммами, но не самого разбиения. Приведем этот алгоритм:

Пока в наборе более одного элемента:

- извлечь из набора два наибольших числа x и y ($x \leq y$),
- поместить в набор разность $y - x$.

Единственное число, которое останется в наборе после выполнения алгоритма, и будет ответом (приближенной разностью).

Рассмотрим работу алгоритма на приведенном выше примере:

- $\{3, 3, 4, 5, 8\}$: извлечь 5 и 8, добавить 3.
- $\{3, 3, 3, 4\}$: извлечь 3 и 4, добавить 1.
- $\{1, 3, 3\}$: извлечь 3 и 3, добавить 0.
- $\{0, 1\}$: извлечь 0 и 1, добавить 1.
- $\{1\}$: ответ равен 1.

Немного подумав, Петя осознал, каким же образом можно на основании этого алгоритма получить и само разбиение с такой разностью. Вам предстоит повторить его подвиг за 5 часов.

Формат входных данных

В первой строке записано целое число n — количество чисел в наборе ($1 \leq n \leq 2 \cdot 10^5$). Во второй строке записаны n целых положительных чисел a_1, a_2, \dots, a_n — числа из набора ($1 \leq a_i \leq 10^9$).

Формат выходных данных

В первой строке выведите d — модуль разности сумм в разбиении, которое получается в соответствии с алгоритмом. Далее вы должны вывести само разбиение. Во второй строке выведите n_1 ($n_1 \geq 0$) — количество чисел в первом наборе. В этой же строке выведите числа, входящие в первый набор (в любом порядке). В третьей строке выведите n_2 ($n_2 \geq 0, n_1 + n_2 = n$) — количество чисел во втором наборе. В этой же строке выведите числа, входящие во второй набор (в любом порядке).

Ваш ответ будет засчитан в том случае, если вы правильно посчитаете значение d в соответствии с алгоритмом, а также предъявите **любое** корректное разбиение, при котором модуль разности сумм в наборах равен d .

Примеры

ввод с клавиатуры	вывод на экран
5 3 3 4 5 8	1 2 4 8 3 3 3 5
4 1 2 3 4	0 2 2 3 2 1 4
5 5 5 4 3 3	2 3 3 3 5 2 4 5

Замечание

Обратите внимание на то, что в третьем примере оптимальным разбиением является $\{5, 5\}$ и $\{4, 3, 3\}$ (суммы обеих частей равны 10). Однако, в соответствии с алгоритмом, который вы должны реализовать, получается разбиение, в котором суммы частей отличаются на 2.

Гарантируется, что не менее чем в 50% тестов жюри $n \leq 2000$.

Задача J. Посылки

Ярослав — владелец небольшой курьерской службы. Недавно он приобрел и внедрил новую систему обработки посылок. Каждая посылка представляет собой коробку, имеющую свой вес и прочность. Система устроена так, что внутри нее находится изначально пустая платформа, на которую можно ставить коробки по следующим правилам:

- Если платформа пуста, коробка ставится непосредственно на платформу, в противном случае — на самую верхнюю коробку, находящуюся на платформе.
- Суммарный вес всех коробок, находящихся на платформе, в любой момент времени не должен превышать прочности платформы S .
- Прочность любой находящейся на платформе коробки в любой момент времени должна быть не меньше суммарного веса коробок, стоящих выше.

Снимать с платформы можно только самую верхнюю коробку.

На вход системе поступают n посылок, при этом i -я посылка поступает ровно в момент времени in_i , ее вес и прочность равны w_i и s_i соответственно. Каждая посылка имеет свою стоимость v_i бурлей, однако, чтобы получить эту стоимость, системе необходимо выдать посылку ровно в момент времени out_i , в противном случае Ярослав получит за нее 0 бурлей. Таким образом, любую посылку можно пропустить и не ставить на платформу, формально выдав ее в момент времени in_i и не получив за нее ничего.

Любая операция в задаче выполняется мгновенно. Это означает, что в один и тот же момент времени можно совершить сразу несколько операций по приему и выдаче посылок, причем в любом порядке.

Обратите внимание, что посылка, выданная в момент времени out_i , сразу же оказывается вне системы, и следующие действия, происходящие в этот же момент времени, производятся без ее учета.

Поскольку система очень сложна, а поступающих посылок очень много, Ярослав просит Вас сказать, какую максимальную сумму денег он может получить с помощью своей системы.

Формат входных данных

В первой строке входных данных содержатся два числа, разделенных пробелом n и S ($1 \leq n \leq 500$, $0 \leq S \leq 1000$). Далее следует n строк, в i -й из этих строк содержатся пять чисел, разделенных пробелом: in_i , out_i , w_i , s_i и v_i ($0 \leq in_i < out_i < 2n$, $0 \leq w_i, s_i \leq 1000$, $1 \leq v_i \leq 10^6$). Гарантируется, что для любых i и j ($i \neq j$) либо $in_i \neq in_j$, либо $out_i \neq out_j$.

Формат выходных данных

Выведите единственное число — максимальную сумму (в бурлях), которую может получить Ярослав.

Примеры

ввод с клавиатуры	вывод на экран
3 2 0 1 1 1 1 1 2 1 1 1 0 2 1 1 1	3
5 5 0 6 1 2 1 1 2 1 1 1 1 3 1 1 1 3 6 2 1 2 4 5 1 1 1	5

Замечание

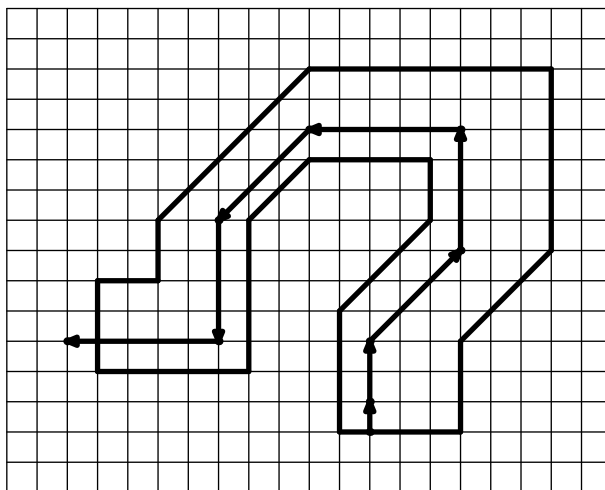
Пояснение ко второму примеру (T — момент времени):

- $T = 0$: Приходит первая посылка, ставим ее на пустую платформу.
- $T = 1$: Приходят вторая и третья посылки, ставим третью на текущую верхнюю на платформе посылку (т.е. первую), затем вторую на третью. Теперь первая посылка держит вес $w_2 + w_3 = 2$, третья — $w_2 = 1$.
- $T = 2$: Выдаем вторую посылку, получая $v_2 = 1$ бурль. Теперь первая посылка держит вес $w_3 = 1$, третья — 0.
- $T = 3$: Приходит четвертая посылка. Сначала выдаем третью посылку, получая $v_3 = 1$ бурль. Теперь первая посылка держит вес 0. Ставим на нее четвертую посылку — первая держит вес $w_4 = 2$.
- $T = 4$: Приходит пятая посылка. Поставить ее на верхнюю на платформе мы не можем, так как первая посылка, в таком случае, будет держать вес $w_4 + w_5 = 3$, что выше ее прочности $s_1 = 2$, что недопустимо. Пропускаем пятую посылку, не получая за нее ничего.
- $T = 5$: Ничего не происходит.
- $T = 6$: Выдаем сначала четвертую, потом первую посылку, получая за них $v_1 + v_4 = 3$ бурля.

Заметим, что можно было пропустить четвертую посылку и поставить пятую вместо нее, однако в этом случае полученная сумма составила бы 4 бурля.

- Перед каждым ходом, **кроме самого первого**, игрок может изменить скорость своей машины не более чем на 1. Скорость не может стать отрицательной. Первый ход, таким образом, выполняется со скоростью 1.
- При перемещении машина последовательно перемещается на выбранное число клеток в выбранном направлении (при этом перемещение по диагонали считается перемещением на одну клетку). На листочке это отмечается отрезком. Этот отрезок не может иметь общих точек с ломаными, ограничивающими трассу (в этом случае считается, что машина врезалась в ограждение и выбывает из игры). Кроме того, этот отрезок не может иметь общих точек со стартовым отрезком: нельзя «вернуться» на стартовый отрезок или проехать через него (исключение из этого правила составляет самое первое перемещение).

Победителем гонки считается тот игрок, траектория машины которого первой пройдет через внутреннюю точку финишного отрезка. Финиш считается корректным, даже если после этого машина нарушает одно из правил передвижения по трассе (траектория пересекает ограничивающие ломаные или линию старта), то финиш считается корректным. Другими словами, главное финишировать, а что после этого будет с машиной, не важно.



Одна из возможных траекторий машины.

Петя очень хочет научиться играть в эту игру как можно лучше. Для этого ему требуется анализировать различные трассы. Ваша задача — написать программу, которая ему в этом поможет. Вам будут заданы ломаные, ограничивающие трассу. Вам нужно будет для каждой стартовой позиции определить минимальное количество ходов, за которое можно довести машину до финиша с этой позиции.

Формат входных данных

В первой строке записано целое число n ($n \geq 2$) — количество вершин в первой ломаной. Каждая из следующих n строк содержит по два целых числа x_i, y_i — координаты вершин ломаной. Первая вершина считается началом ломаной, последняя — концом ломаной. Далее в том же формате описана вторая ломаная. В описании каждой ломаной не встречается трех или более подряд идущих точек, лежащих на одной прямой.

Гарантируется, что ломаные удовлетворяют всем ограничениям, описанным в условии. Координаты вершин ломаных — от 0 до 50.

Гарантируется, что стартовый и финишный отрезки имеют длину не менее 2.

Формат выходных данных

В первой строке выведите k — число стартовых позиций на трассе. Во второй строке выведите ответы для этих позиций. В том случае, если стартовый отрезок горизонтальный, выводите ответы

слева направо. Если же отрезок вертикальный, то снизу вверх. Если для каких-то стартовых позиций не существует способа доехать до финиша, выведите для таких позиций ответ -1.

Примеры

ввод с клавиатуры	вывод на экран
8 15 2 15 5 18 8 18 14 10 14 5 9 5 7 3 7 8 11 2 11 6 14 9 14 11 10 11 8 9 8 4 3 4	3 8 9 8
4 6 0 6 3 4 5 0 5 4 9 0 9 2 4 7 0 7	2 4 5
4 0 0 2 0 2 2 3 2 4 0 2 1 2 1 4 3 4	1 -1

Замечание

Первый пример соответствует картинке из условия.