

Задача А. Игра с мячом

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Воспитательница детского сада Наталья Павловна придумала новую развивающую игру с мячом. Эта игра не только развивает детишек в плане физического развития, но при этом еще обучает их счету.

Игра заключается в следующем. Детишки стоят по кругу. Условимся считать, что они пронумерованы числами от 1 до n по часовой стрелке и ребенок с номером 1 держит мяч. Вначале первый ребенок кидает мяч следующему за ним по кругу, т.е. ребенку с номером 2. Дальше ребенок с номером 2 кидает мяч ребенку, стоящему через одного, т.е. с номером 4, затем четвертый ребенок кидает мяч ребенку, стоящему через 2 от него, т.е. ребенку с номером 7, дальше мяч достается ребенку, стоящему через 3 от очередного, дальше — через 4 и т.д. Заметим, что при броске мяча может произойти переход через начало круга. Например, если $n = 5$, то после третьего броска мяч достанется снова ребенку с номером 2. Всего делается $n - 1$ бросок, и игра заканчивается.

Проблема заключается в том, что не всем детям в процессе игры достается мяч. Если ребенку не достается мяч, он очень расстраивается и плачет до тех пор, пока Наталья Павловна не даст ему конфетку. Поэтому Наталья Павловна просит вас помочь ей определить номера детишек, которым будет доставаться мяч после каждого броска.

Формат входного файла

В первой строке содержится целое число n ($2 \leq n \leq 100$) — количество детишек в кругу.

Формат выходного файла

В единственной строке выведите $n - 1$ число — номера детишек, которым будет доставаться мяч после каждого броска. Числа разделяйте пробелами.

Примеры

<code>stdin</code>	<code>stdout</code>
10	2 4 7 1 6 2 9 7 6
3	2 1

Задача В. Футболки от спонсора

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Один известный спонсор одного известного соревнования решил одарить каждого участника соревнования футболкой. Возникла понятная проблема: с одной стороны, непонятно, сколько каких размеров нужно запастись футболок, а с другой стороны, много футболок заказывать не хочется (не на стены же их потом клеить). Поломав голову и сделав предварительные прикидки, представители спонсоров заказали некоторое количество футболок размеров S, M, L, XL и XXL. Футболки получились удачными, поэтому в день соревнования за ними выстроилась очередь из K участников. Каждый участник характеризуется своим желаемым размером футболки (так получилось, что для всех участников это тоже один из размеров S, M, L, XL и XXL). Участники по очереди подходят за футболкой и стараются выбрать себе самую подходящую, выбирая ее следующим образом. Если еще осталась футболка желаемого размера, то он берет ее, не раздумывая. В противном случае для него предпочтительнее взять футболку с размером, как можно более близким к желаемому (расстояние между соседними размерами считается равным единице). В случае, если выбор неоднозначен, участник возьмет футболку большего размера (навырост). К примеру, для человека с размером L список предпочтений выглядит так: L, XL, M, XXL, S. По данным о том, сколько футболок какого размера запасли организаторы, и о размерах участников в очереди, определите, кому досталась футболка какого размера.

Формат входного файла

В первой строке записано пять неотрицательных целых чисел $N_S, N_M, N_L, N_{XL}, N_{XXL}$, не превосходящих 1000 — количества футболок соответствующих размеров. Во второй строке записано целое число K ($1 \leq K \leq 1000$) — количество участников. Следующие K строк содержат желаемые размеры футболок для участников. Размеры даны в том порядке, в каком участники выстроились в очередь. Гарантируется, что $N_S + N_M + N_L + N_{XL} + N_{XXL} \geq K$.

Формат выходного файла

Для каждого участника в отдельной строке выведите размер футболки, которую он получил.

Примеры

stdin	stdout
1 0 2 0 1	XXL
3	L
XL	L
XXL	
M	

Задача С. Хомячки и тигры

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Сегодня в цирке необычное представление — на арене одновременно выступают хомячки и тигры! Все они выстроились в круг по краю арены, и теперь перед дрессировщиком стоит непростая задача: он хочет переставить животных так, чтобы все хомячки стояли подряд и все тигры тоже стояли подряд. Чтобы не создавать столпотворения, дрессировщик меняет животных парами. Он подает команду двум животным выйти из круга и поменяться местами. Поскольку хомячки в присутствии тигров ощущают себя крайне некомфортно, да и тигры нервничают в присутствии большого количества потенциальной добычи (состоящей не только из хомячков, но и из более аппетитных зрителей), дрессировщик хочет справиться с перестановкой животных как можно быстрее, т.е. за наименьшее количество обменов. Ваша задача — помочь ему.

Формат входного файла

В первой строке содержится число n ($2 \leq n \leq 1000$) — общее количество животных на арене. Во второй строке содержится описание расположения животных. Строка состоит из n символов «Н» и «Т». Символы «Н» соответствуют хомячкам, символы «Т» — тиграм. Гарантируется, что на арене присутствует хотя бы один хомячок и хотя бы один тигр. Животные заданы в порядке расположения по кругу, при этом последнее животное стоит рядом с первым.

Формат выходного файла

Выведите единственное число — наименьшее количество обменов, позволяющее дрессировщику достичь своей цели.

Примеры

<code>stdin</code>	<code>stdout</code>
3 НТН	0
9 НТНТНТНТНТ	2

Note

В первом примере никого не нужно менять местами, потому что животные каждого вида и так стоят подряд. Во втором примере можно поменять, например, тигра в позиции 2 с хомячком в позиции 5 и затем — тигра в позиции 9 с хомячком в позиции 7.

Задача D. Парковка

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

В наши дни становится все труднее удачно припарковать свой автомобиль в городе. Представим себе отрезок улицы длиной в L метров, вдоль которого разрешена парковка. Водители должны ставить свои автомобили строго параллельно тротуару с правой стороны улицы (помните, что в стране авторов задач движение правостороннее!). Каждый водитель при парковке хочет оставить себе некоторый запас для маневрирования, поэтому водитель ищет такое место, что расстояние до сзади стоящей машины будет не меньше b метров, а до впереди стоящей — не меньше f метров (если сзади машины нет, то автомобиль можно ставить у самого края отрезка для парковки; то же самое верно для случая, когда спереди не припарковано машин). Введем ось координат вдоль тротуара. Пусть зона для парковки начинается в точке 0 и заканчивается в точке L . Водители едут в направлении увеличения координаты и ищут самое раннее место (с наименьшей возможной координатой), где они могут припарковать машину. В случае, если такого места не находится, водитель едет дальше в поисках счастья. Иногда некоторые машины покидают улицу и освобождают место для парковки. Считая, что на улице никогда не находятся две двигающиеся машины одновременно, напишите программу, которая по информации о водителях, приезжающих на улицу в целях парковки и уезжающих с нее, моделирует процесс и определяет для каждого автомобиля место, занятое им в зоне парковки.

Формат входного файла

В первой строке записаны три целых числа L , b и f ($10 \leq L \leq 100000$, $1 \leq b, f \leq 100$). Во второй строке дано целое число n ($1 \leq n \leq 100$) — количество запросов к вашей программе. Каждый запрос описывается в отдельной строке и задается двумя числами. Первое число означает тип запроса. Если тип запроса равен 1, то в этом случае второе число означает длину очередного автомобиля (в метрах), который въезжает на улицу в поисках места для парковки. Если же тип запроса равен 2, то тогда второе число означает номер запроса (начиная с 1) такого, что автомобиль, чей приезд на парковку описывался запросом с этим номером, ее покидает. Гарантируется, что этот автомобиль в момент запроса типа 2 был припаркован. Длины автомобилей — это целые числа от 1 до 1000.

Формат выходного файла

Для каждого запроса типа 1 в отдельной строке выведите число -1 , если соответствующему автомобилю не нашлось места на парковке вдоль улицы. В противном случае выведите целое число, равное расстоянию от задней части автомобиля в его припаркованном положении до начала зоны парковки.

Примеры

stdin	stdout
30 1 2 6 1 5 1 4 1 5 2 2 1 5 1 4	0 6 11 17 23
30 1 1 6 1 5 1 4 1 5 2 2 1 5 1 4	0 6 11 17 6
10 1 1 1 1 12	-1

Задача E. Расческа

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Преодолев все испытания, Лара Крофт наконец-то оказалась в комнате с сокровищами. К своему удивлению, не найдя там горы золота, Лара осмотрелась и заметила, что на полу нарисована таблица $n \times m$ из плит, на которых написаны целые числа, а у стены валяется огромное число камней. На колонне у таблицы Лара нашла поясняющую табличку, которая гласила, что для получения сокровищ надо в каждой строке таблицы выбрать некоторое ненулевое количество первых плит и положить камни на все эти плиты, придавив их. После этого она получит число золотых монет, равное сумме чисел, написанных на выбранных плитах. Лара быстро придумала, как разложить камни, и уже собиралась сделать это, как заметила приписку к таблице, сделанную внизу и мелким шрифтом. Согласно этой приписке, для того, чтобы своды комнаты не обвалились и не придавили искателя приключений, выбранные плиты должны образовывать расческу. Пояснилось, что выбранные плиты образуют расческу, если последовательность c_1, c_2, \dots, c_n , составленная из количеств плит, выбранных в каждой строке таблицы, удовлетворяет следующему свойству: $c_1 > c_2 < c_3 > c_4 < \dots$, т.е. знак неравенства между соседними элементами чередуется. Теперь Лара в растерянности и не знает, что делать. Помогите ей определить наибольшее количество монет, которое она может получить, выжив при этом.

Формат входного файла

В первой строке записана пара целых чисел n, m ($2 \leq n, m \leq 1500$). Следующие n строк содержат по m целых чисел каждая — сама таблица. Числа в таблице по модулю не превосходят 10000.

Формат выходного файла

Выведите единственное число — наибольшее количество монет, которое Лара может получить.

Примеры

stdin	stdout
2 2 -1 2 1 3	2

Задача F. Задача Эркюля Пуаро

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	256 megabytes

Сегодня вам предстоит решить задачу, с которой не может справиться сам Эркюль Пуаро! Именно поэтому данное преступление до сих пор не раскрыто, и эта история так и не вошла в детективы Агаты Кристи.

Вам не сообщается, какое совершено преступление, где и когда обнаружен труп и остальные детали. Известно только то, что преступление совершено в доме, имеющем n комнат и m дверей между парами комнат. Жильцы дома очень подозрительны, поэтому все двери запираются на ключ и все ключи различны. Согласно показаниям свидетелей в ночь с четверга на пятницу все двери в доме были заперты и известно, в каких комнатах находились жильцы и какие ключи у кого были. Аналогичная информация известна и про ночь с пятницы на субботу, в которую все двери также были заперты. В пятницу лил сильный дождь, поэтому из дома никто не выходил и никто в него не входил. В течение дня обитатели дома могли

- открывать и закрывать двери в соседние комнаты при помощи имеющихся ключей (каждую дверь можно открывать и закрывать с любой стороны);
- свободно перемещаться между комнатами, если соответствующие двери открыты;
- передавать друг другу ключи, находясь в одной комнате.

«Серые клеточки» Эркюля Пуаро не способны справиться с таким количеством информации. Определите, могло ли из расположения людей и ключей в первую ночь получиться расположение во вторую ночь, или кто-то из свидетелей точно лжет.

Формат входного файла

В первой строке заданы три числа n , m и k ($1 \leq n, m, k \leq 1000$) — количество комнат, количество дверей и количество жильцов в доме соответственно. Следующие m строк содержат пары номеров комнат, которые соединяют двери. Комнаты нумеруются целыми числами от 1 до n . Между парой комнат может быть более одной двери. Никакая дверь не соединяет комнату саму с собой. Следующие k строк описывают расположение жильцов в первую ночь. В каждой строке содержится имя жильца (непустая строка из не более 10 латинских букв), через пробел следует номер комнаты, затем через пробел — количество имеющихся у жильца ключей. Далее через пробел перечислены номера дверей, которые открываются имеющимися ключами. Двери нумеруются целыми числами от 1 до m в порядке их описания во входных данных. Все имена жильцов различны, строчные и заглавные буквы различаются. Каждый из m ключей встречается в описании ровно один раз. Несколько людей могут находиться в одной комнате, некоторые комнаты могут быть пусты. Следующие k строк описывают расположение жильцов во вторую ночь точно в таком же формате. Гарантируется, что в описании второй ночи имена жильцов те же самые и каждый из m ключей встречается ровно один раз.

Формат выходного файла

Выведите «YES» (без кавычек), если из первого расположения можно получить второе, и «NO» в противном случае.

Примеры

stdin	stdout
2 1 2 1 2 Dmitry 1 1 1 Natalia 2 0 Natalia 1 1 1 Dmitry 2 0	YES
4 4 3 1 3 1 2 2 3 3 4 Artem 1 1 4 Dmitry 1 1 2 Edvard 4 2 1 3 Artem 2 0 Dmitry 1 0 Edvard 4 4 1 2 3 4	NO

Задача G. Задача Императора

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Это произошло во времена Великой Берляндской Империи. Однажды Император увидел сон, в котором к нему явился Посланник богов и приказал построить храм с выпуклым n -угольником в основании. На следующее утро Император отдал распоряжение возвести храм, в основании которого лежит правильный n -угольник. Храм был построен, но вскоре Империю постигли бедствия и неурожая. После того, как случилось землетрясение, разрушившее храм, Император понял, что чем-то навлек на свой народ немилость богов. Он приказал позвать к себе Мудреца. Когда явился Мудрец, Император поведал ему о своем сне и задал вопрос: «Дай же мне ответ, о Мудрейший из Мудрых, чем мог я прогневить богов?». «О, Владыко!» — отвечал мудрец. — «Насколько дано мне судить, боги гnevаются оттого, что ты поторопился выполнять их поручение и не дослушал Посланника до конца».

Действительно, на следующую же ночь Посланник явился вновь. Он укорял Императора за то, что тот выбрал несовершенную форму для храма. «Так какая же форма может быть совершеннее, чем правильный n -угольник?!» — воскликнул император во сне. На что Посланник дал подробный и обстоятельный ответ.

- Все вершины многоугольника должны располагаться в точках с целочисленными координатами.
- Все длины его сторон должны быть различны.
- Из всех таких многоугольников должен быть выбран такой, длина максимальной стороны которого минимальна.

Вы — подневольный архитектор, которому предстоит спроектировать храм. Учтите, что n -угольник должен быть простым (имеющим границу без самопересечений и наложений) и выпуклым, но при этом допускается, чтобы три последовательные вершины лежали на одной прямой.

Формат входного файла

В первой строке задано единственное число n ($3 \leq n \leq 10000$).

Формат выходного файла

В первой строке выведите «YES» (без кавычек), если n -угольник, обладающий требуемыми свойствами, построить возможно. В следующих n строках выведите целочисленные координаты вершин многоугольника в порядке обхода против часовой стрелки. Координаты не должны превосходить 10^9 по абсолютной величине. Никакие две вершины не должны совпадать. Разрешается выводить любое из возможных решений. Если многоугольник построить невозможно, выведите «NO».

Примеры

stdin	stdout
3	YES 0 0 1 0 0 2
3	YES 0 1 -1 0 -1 -1