

Задача А. Арифметическая магия

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Дэвид Блейн попросил зрителя задумать два числа. Затем он попросил перемножить два числа, большие каждого из задуманных на единицу, вычесть из результата сначала одно задуманное число, затем другое, а затем — их произведение, а полученный результат возвести в N -ю степень.

После чего Дэвид внимательно взгляделся в лицо зрителя и правильно назвал получившийся результат. Ваша задача — повторить фокус Дэвида. По заданному N угадайте получившееся у зрителя число.

Формат входных данных

Входные данные содержат одно целое число N ($0 \leq N \leq 1000$).

Формат выходных данных

Выведите одно число — получившийся у зрителя результат.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	1

Задача В. Большие переменные

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

В Тридевятом царстве N городов. Король хочет соединить города $N - 1$ двусторонними авиа-линиями так, чтобы:

- Каждая авиалиния соединяла два различных города.
- Из каждого города можно было добраться до любого другого напрямую или с пересадками.

Назовём *доступностью* города количество городов, с которыми он напрямую соединён авиалиниями. Король также требует, чтобы **максимальное** значение доступности среди всех городов королевства было как можно **большим**.

Сколько существует различных способов соединить города требуемым образом?

Формат входных данных

Входные данные содержат одно целое число N — количество городов ($2 \leq N \leq 10^9$).

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	3

Замечание

Поясним пример к задаче. Максимальное возможное значение доступности для трёх городов равно 2. Возможны три различных варианта соединения, при котором оно достигается: $(1,2)(2,3)$, $(1,3)(2,3)$, $(1,2)(1,3)$.

Задача С. Совпадения

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Участников ICPC (Intergalactic Collegiate Programming Contest) поселили в только что построенную гостиницу. Всего в гостинице N одноместных комнат, занумерованных целыми числами от 1 до N без пропусков. Для каждого участника известен номер его паспорта — целое число от 1 до 10^9 включительно. Номера паспортов у участников с разных планет могут совпадать.

Ожидая заселения, несколько участников заметили, что возможна забавная ситуация: номер паспорта может совпасть с номером комнаты. После чего участники задались вопросом — а какое наибольшее число таких совпадений было бы возможно, если специально расселять участников так, чтобы максимизировать это значение?

По заданному количеству комнат в гостинице и списку номеров паспортов участников найдите ответ на этот вопрос.

Формат входных данных

Первая строка входных данных содержит одно целое число N ($1 \leq N \leq 10^5$). i -я из последующих N строк содержит целое число a_i — номера паспорта i -го участника ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — наибольшее количество совпадений номеров паспортов с номерами комнат, которое возможно получить при расселении участников по комнатам.

Примеры

стандартный ввод	стандартный вывод
5 1 3 5 7 5	3
4 1000000000 1000000000 1000000000 1000000000	0

Задача D. Драфт НБА

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Перед каждым сезоном НБА проходит драфт, то есть церемония выбора игроков командами.

Перед драфтом скаутские службы команд детально изучают потенциальных новичков. Одна из команд поручила вам написать программу для обработки собранных данных в соответствии с внутриклубными стандартами.

Каждый из новичков до этого провёл как минимум один сезон в студенческих лигах, так что для каждого игрока известны пять основных целочисленных параметров:

- Рост игрока — ожидаемый диапазон от 190 до 220 см.
- Размах рук (иначе говоря, wingspan) — ожидаемый диапазон от 200 до 250 см.
- Среднее количество очков за матч — ожидаемый диапазон от 10 до 20.
- Среднее количество подборов за матч — ожидаемый диапазон от 2 до 6.
- Среднее количество передач за матч — ожидаемый диапазон от 3 до 7.

Таким образом, каждый из параметров может оказаться ниже ожидаемого диапазона, в нижней половине ожидаемого диапазона, в верхней половине ожидаемого диапазона (середина входит в верхнюю половину) и выше ожидаемого диапазона.

Перед драфтом требуется распределить игроков по следующим категориям:

- уникальный игрок (таких ещё называют «единорогами» — unicorn) с сочетанием выдающихся физических данных и игровых навыков;
- игрок, достойный выбора в первом раунде;
- игрок, достойный выбора во втором раунде;
- игрок, которого не стоит выбирать вообще.

Если у игрока три параметра выше ожидаемого диапазона, **причём среди них обязательно есть рост или размах рук**, то игрок считается «единорогом» (категория 0)

Игрока рекомендуется выбирать в первом раунде драфта (категория 1), если верно одно из следующих утверждений:

- У игрока два параметра выше ожидаемого диапазона и ещё один — как минимум в верхней половине ожидаемого диапазона.
- У игрока все параметры как минимум в ожидаемом диапазоне и не менее трёх — как минимум в верхней половине.

Игрока рекомендуется выбирать во втором раунде драфта (категория 2), если верно одно из следующих утверждений:

- У игрока один параметр выше ожидаемого диапазона и ещё один — как минимум в верхней половине ожидаемого диапазона.
- У игрока три параметра как минимум в верхней половине ожидаемого диапазона.

В остальных случаях тратить выбор драфта на этого игрока не рекомендуется (категория 3).

Вам заданы параметры некоторых из 32 игроков текущего драфта (для объективности имена игроков не указаны). Ваша задача — для каждого игрока определить категорию, под которую он подпадает для вашей команды на драфте.

Формат входных данных

Первая строка входных данных содержит одно целое число N — количество игроков ($1 \leq N \leq 32$). Описание каждого игрока состоит из 5 целых чисел: роста h ($150 \leq h \leq 250$), размаха рук w ($150 \leq w \leq 270$), среднего количества очков s за матч ($5 \leq s \leq 35$), среднего количества подборов r за матч ($0 \leq r \leq 10$), среднего количества передач p за матч ($0 \leq p \leq 10$). Каждое число задаётся в отдельной строке.

Формат выходных данных

Для каждого игрока в порядке их следования во входном файле выведите одно число — категорию на драфте, в которую этот игрок попадает в соответствии с критериями вашей команды.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	0
230	2
190	1
16	
7	
9	
205	
225	
15	
5	
2	
210	
210	
30	
9	
9	

Замечание

У первого игрока из примера к задаче рост выше ожидаемого диапазона, кроме того, выше ожидаемого ещё два параметра. Так что этот игрок является «единорогом». Второму игроку для выбора в первом раунде не хватает количества передач — оно у него ниже ожидаемого диапазона, а все остальные навыки в верхней половине этого диапазона. Так что команда будет выбирать этого игрока во втором раунде. У третьего игрока три параметра выше ожидаемого диапазона, но так как среди них нет ни роста, ни размаха рук, «единорогом» этот игрок не является и будет выбран в первом раунде.

Задача Е. Думский регламент

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

В Тридевятом Царстве уже много лет как установилась конституционная монархия. В парламент Тридевятого царства входят 26 партий, обозначаемых строчными буквами английского алфавита от 'a' до 'z'. Заседание парламента в соответствии с регламентом проходит по следующей схеме:

- В начале заседания никаких законопроектов на повестке дня не стоит.
- Любая партия (в том числе и та, которая внесла текущий законопроект) в любой момент времени может внести на обсуждение новый законопроект.
- При вносе партией нового законопроекта обсуждение предыдущего приостанавливается и начинается обсуждение нового. После того, как парламентарии голосуют за новый законопроект, обсуждение законопроекта, прерванного им, продолжается.
- Дискуссия по любому законопроекту обязана завершиться голосованием по этому законопроекту.
- В конце заседания не должно остаться внесённых законопроектов, дискуссия по которым не завершена.

Вам дана запись о событиях на сессии парламента. Каждое событие является либо внесением нового законопроекта, либо голосованием за какой-то законопроект, причём в обоих случаях известен номер партии, этот законопроект предложившей. События даны в том порядке, в котором они происходили. Проверьте, может ли данная запись о событиях соответствовать какому-либо заседанию, удовлетворяющему порядку проведения, описанному выше.

Формат входных данных

Первая строка входных данных содержит одно целое число K — количество строк в записи сессии ($1 \leq K \leq 1000$). Каждая строка соответствует одному из двух событий:

- **Add x** — партия x внесла на голосование законопроект.
- **Vote x** — прошло голосование за документ, внесённый партией x .

Здесь x — строчная буква английского алфавита от 'a' до 'z', задающая партию.

Формат выходных данных

Выведите «Yes», если существует корректный порядок проведения заседания, который мог привести к такой записи, и «No», если ни при каком корректном порядке проведения заседания данная запись появиться не могла.

Примеры

стандартный ввод	стандартный вывод
4 Add a Add b Vote a Vote b	No
8 Add z Vote z Add x Add y Add x Vote x Vote y Vote x	Yes
1 Vote z	No

Задача F. Правильный подмногоугольник

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Задан правильный N -угольник. Требуется выбрать наименьшее количество его вершин, которые также образуют правильный многоугольник.

Формат входных данных

Входные данные содержат одно целое число N ($3 \leq N \leq 10^{12}$).

Формат выходных данных

Выведите одно число — наименьшее количество вершин заданного многоугольника, которые образуют правильный многоугольник.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5	5
21	3

Задача G. Есть ли делитель?

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

На форуме, на котором обсуждаются задачи олимпиад по информатике, ввели следующий аналог капчи. Участнику выдаётся строка из N десятичных цифр (без ведущих нулей). В качестве ответа требуется ввести такое основание системы счисления B , что в этой системе счисления выданная запись будет соответствовать составному числу (назовем его D), а также число X , большее 1 и меньшее D , являющееся делителем D .

При этом B и X не должны превосходить 10^9 .

По заданной строке десятичных цифр найдите любую пару чисел B и X , удовлетворяющую ограничениям, или ответьте, что решения в заданных ограничениях не существует.

Формат входных данных

Входные данные состоят из непустой строки длиной до $3 \cdot 10^6$ символов, составленной из цифр от 0 до 9 и не начинающейся с 0.

Формат выходных данных

Если решение существует, выведите два числа — основание системы счисления B и делитель X , записанные в десятичной системе счисления. Оба числа должны удовлетворять ограничениям $2 \leq B, X \leq 10^9$. Если решения не существует, выведите -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	-1
4	10 2
19	11 2

Задача Н. ICPC

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Словарь тау-китянского языка состоит из всех слов, длина которых не превосходит N . Слова записываются строчными буквами английского алфавита. В тау-китянском словаре — в отличие от земных — слова сначала упорядочены по длине, а только затем по алфавиту, то есть сначала идут однобуквенные слова от 'a' до 'z', затем двухбуквенные — от 'aa' до 'zz' и так далее.

Алиса выписала подряд все слова тау-китянского языка в том порядке, в котором они перечислены в словаре, и получила длинное слово. Сколько раз в этом слове встретится подстрока «icpc»?

Формат входных данных

Входные данные содержат одно целое число N ($1 \leq N \leq 10^9$) — максимальная длина слова в тау-китянском языке.

Формат выходных данных

Выведите одно число — остаток от деления количества вхождений подстроки «icpc» на $10^9 + 7$.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	0
5	134

Задача I. Составление задач

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Для составления задач на очередное соревнование организаторы пригласили P опытных участников этих же соревнований. Каждый из них знает некоторый набор задач. Участники любят делиться идеями задач, поэтому каждая задача может быть известна нескольким участникам. К сожалению, если участник знает хотя бы одну задачу из выбранного на соревнование комплекта, то он не может принять в нём участие.

Организаторы заинтересованы в массовости соревнования, поэтому они хотят выбрать задачи так, чтобы как можно больше участников могли принять в нём участие. При этом набор задач должен быть не пуст, и, по возможности, как можно больше.

Организаторы не являются программистами, поэтому испытывают проблемы с тем, чтобы набрать задачи требуемым образом, поэтому просят вас помочь им.

Формат входных данных

В первой строке входных данных записаны три целых числа P , T и M ($1 \leq P, T \leq 10^5$; $0 \leq M \leq \min(10^6, P \cdot T)$) — количество участников, задач и пар участник-задача, которую он знает, соответственно.

В следующих M строках описаны задачи, которые известны участникам. В каждой строке записаны два целых числа u и v ($1 \leq u \leq P$; $1 \leq v \leq T$) — номер участника и номер одной из известных ему задач. Гарантируется, что для каждого участника каждая известная ему задача описана ровно один раз.

Формат выходных данных

В первой строке выведите два числа P_0 и T_0 — искомое количество участников и задач. Обратите внимание — в первую очередь требуется максимизировать количество участников, а при максимальном числе участников — количество задач.

Во второй строке выведите T_0 чисел — номера задач в набранном комплекте. Все номера должны быть различными натуральными числами, не превосходящими T . Если возможно несколько оптимальных решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
3 4 6 1 1 1 2 2 2 2 3 3 3 3 4	2 1 1
3 5 6 1 1 1 2 2 1 2 3 3 1 3 3	3 2 4 5

Замечание

В первом примере можно было также взять задачу с номером 4, поскольку ее знает один человек. Во втором примере никто не знает задач с номерами 4 и 5, поэтому лучше всего взять их.

Обратите внимание — в этой задаче требуется считать из входного потока много данных. Для этого в программе на языке C++ следует добавить в начало функции `main` следующие три строчки:

```
ios_base::sync_with_stdio();  
cin.tie(0);  
cout.tie(0);
```

В программе на языке **Java** следует отказаться от использования класса **Scanner**.

Задача J. Порталы

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Вы очнулись в тёмной комнате и сразу поняли, что оказались в лабиринте, из которого надо выбраться! Лабиринт представляет собой прямоугольник из N строк по M клеток в каждой, некоторые из которых заняты стеклянными или сплошными стенами, причем весь лабиринт окружают сплошные стены. Если смотреть на карту лабиринта, то строки пронумерованы от 1 до N сверху вниз, а столбцы пронумерованы от 1 до M слева направо. Вы находитесь в одной из не занятых стеной клеток лабиринта, а в какой-то свободной клетке (возможно, в стартовой) находится выход.

Немного побродив по комнате, вы обнаружили порталную пушку, которая предназначена для установки на стене порталов одного из двух цветов — оранжевого или синего. В каждый момент времени может быть не более одного портала каждого цвета, поэтому, когда устройство выставляет новый портал, старый портал того же цвета исчезает. Изначально порталов нет. Стоя в клетке, вы можете выстрелить порталной пушкой в одном из четырех направлений, при этом выстрел пролетает сквозь стеклянные стены и портал создается на ближайшей в этом направлении стороне сплошной стены. Если вы попытаетесь поставить портал на стороне стены, где уже стоит какой-то портал, то ничего не изменится, но будет считаться, что порталная пушка сработала.

Также вы можете передвигаться по лабиринту по любым свободным клеткам, двигаясь в одном из четырёх направлений и входить в находящийся рядом с вами портал — при этом он должен находиться на видимой вами стороне стены в соседней с вами клетке. В таком случае вы мгновенно оказываетесь возле второго портала. Если выход из этого портала ведёт в стеклянную стену, а такое могло случиться, если сплошная и стеклянная стена были рядом, и выстрел порталной пушки пролетел сквозь стеклянную стену и остался на сплошной, то вы погибнете.

Вы опасаетесь, что порталная пушка может выйти из строя, поэтому хотите пройти лабиринт, используя ее как можно меньше раз. Напишите программу, которая рассчитает минимально возможное количество использований устройства, позволяющее вам выйти из лабиринта.

Формат входных данных

В первой строке входных данных через пробел записаны два целых числа N и M ($3 \leq N, M \leq 1000$) — число строк и столбцов клеток в лабиринте. В каждой из следующих N строк записаны M символов, описывающих лабиринт. Символ «W» описывает сплошную стену, символ «G» — стеклянную, а символ «.» — свободную клетку. Гарантируется, что все граничные клетки заняты сплошной стеной.

В следующих двух строках записаны по два числа r_S, c_S, r_E, c_E ($2 \leq r_S, r_E \leq N - 1$; $2 \leq c_S, c_E \leq M - 1$) — номера строк и столбцов стартовой клетки, и клетки с выходом соответственно. Гарантируется, что эти клетки являются свободными. Нумерация клеток начинается с единицы с левого верхнего угла.

Формат выходных данных

Если добрать до выхода невозможно даже используя порталную пушку, выведите только «-1» (без кавычек). Иначе, в первой строке выведите два числа P и S — количество выстрелов порталной пушки и число пройденных шагов соответственно. Вам требуется минимизировать только число P , количество шагов не должно превосходить $2 \cdot n \cdot m$. В следующих $P + S$ строках выведите последовательность ваших действий. Действие описывается двумя символами, типом действия («M», «O» или «B») и его направлением («U», «D», «R» или «L»), выведенными подряд. Типы действий:

- «M» — передвинуться на одну клетку в выбранном направлении. Вы не можете идти в клетку, в которой находится стена, на которой с вашей стороны нет портала.
- «O» — поставить оранжевый портал на ближайшей сплошной стене в выбранном направлении.
- «B» — поставить синий портал на ближайшей сплошной стене в выбранном направлении.

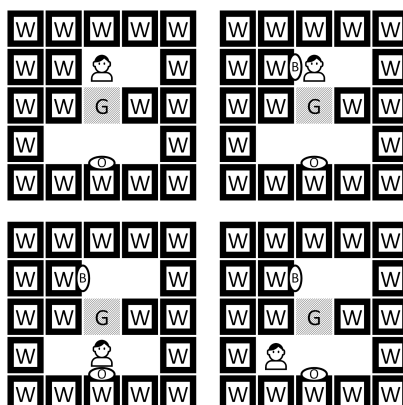
Направление может принимать значения «U», «D», «R» или «L» — вверх, вниз, вправо и влево соответственно.

Примеры

стандартный ввод	стандартный вывод
5 5 WWWWW WW..W WWGWW W...W WWWWW 2 3 4 2	2 2 OD BL ML ML
7 6 WWWWW W..W.W W.W..W W.W..W W.WG.W W...WW WWWWW 2 3 2 5	2 17 ML MD MD MD MD MR MR OU ML ML MU MU MU MU MR BR MR MR MU
5 5 WWWWW W.G.W WW.GW W.G.W WWWWW 2 2 4 2	4 3 OR BU MU BD MR OL MD

Замечание

Иллюстрацию к первому примеру можно увидеть ниже.



Задача К. «Контакт» для двоих

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

Лось Валера очень любит играть в «Контакт» — широко популярную словесную игру; однако поиграть в нее в последнее время удается нечасто, ведь в игре должны участвовать не менее трех игроков, а собраться вместе как минимум вдвоем из-за нехватки времени и нестыкующихся графиков крайне затруднительно...

В силу этого, лось Валера придумал новую игру — «Контакт для двоих». Суть ее заключается в следующем. Один из игроков загадывает некоторое непустое слово S , которое второй игрок должен отгадать, и сообщает первую букву этого слова. Второй игрок, пытаясь угадать слово, называет одно за другим слова, начинающиеся на заданную букву, а первый игрок сообщает ему, верно ли слово, причем второй игрок не должен повторяться. Если второй игрок назвал некоторое наперед заданное число K слов и так и не угадал слово, то первый игрок сообщает вторую букву слова; после этого, второй игрок может называть лишь слова, начинающиеся с этих двух букв и не названные им ранее (в том числе и когда он знал лишь одну букву). Если после появления второй буквы, второй игрок сумеет назвать еще K слов, то первый игрок сообщает третью букву, и т.д. Если в какой-то момент оказывается, что первый игрок уже назвал все буквы своего слова, а второй после этого назвал K слов, начинающихся с него, но само слово не назвал, то первый игрок просто сообщает, что уже открытые буквы и образуют загаданное слово, и игра заканчивается.

Чтобы игра была интереснее, лось Валера, играя за первого игрока, старается загадать такое слово, чтобы второй игрок сделал как можно больше попыток угадать его (удачная попытка также считается). Однако сам процесс игры зависит от словарного запаса второго игрока, а также от выбранного числа K . Так как лось Валера хорошо знает своих друзей, то для каждого из них Валера может выписать все слова, которые тот знает. Конечно же, чтобы не расстраивать второго игрока, Валера всегда загадывает слово, которое игрок знает. Однако, не так-то просто выбрать это слово и число K . Поэтому Валера решил написать программу, которая по заданному словарю и Q парам «загаданное слово – значение K » определяет, какое максимальное число слов может назвать второй игрок, пока игра не закончится угадыванием слова вторым игроком или сообщением слова первым.

Лось Валера справился с задачей. Справитесь ли вы?

Формат входных данных

В первой строке записано единственное целое положительное число N — количество слов в словаре потенциального второго игрока. В каждой из следующих N строк записано одно непустое слово этого словаря, состоящее из строчных букв английского алфавита. Суммарная длина слов не превосходит $2 \cdot 10^5$.

Обратите внимание: слова **могут быть одинаковыми** по написанию, например, потому, что они различаются при произношении ударениями: к примеру, в русском языке слова «замОк» и «зАмок» разные слова, хотя пишутся одинаково. В языке, на котором лось Валера и его друзья любят играть в «Контакт», количество возможных акцентов, делающих одинаково записываемые слова разными по звучанию, очень велико. Во время игры участники полностью различают одинаково записываемые слова, произнося их по-разному.

В $(n + 2)$ -й строке содержится целое число Q , ($1 \leq Q \leq 2 \cdot 10^5$) — количество исследуемых пар «загаданное слово – значение K ». В следующих Q строках записаны по два целых числа w и k ($1 \leq w, k \leq n$) — номер загаданного слова из словаря и предполагаемое значение K . Слова нумеруются в том порядке, в котором они заданы выше, начиная с единицы.

Формат выходных данных

Для каждой из Q пар выведите в отдельной строке единственное целое число — максимальное количество слов, которые может назвать второй игрок.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
6 asassin assistant astronaut abrakadabra abbey automaton 9 1 1 1 2 1 3 4 1 4 2 4 3 6 1 6 2 6 3	3 5 6 3 4 5 2 3 4
3 aa ab ab 6 1 1 2 1 1 2 3 2 2 2 3 1	2 2 3 3 3 2
7 pit pitbul piter pitstop pitlane petroleum pistol 6 1 2 1 3 6 4 7 2 7 3 5 1	6 7 5 5 7 4

Задача L. Ближайшие точки

Ограничение по времени: 2 секунды Ограничение по памяти: 512 мегабайт

На прямоугольной декартовой плоскости задан прямоугольник A с вершинами в точках $(0, 0)$ и (X, Y) , стороны которого параллельны осям координат, где X, Y — целые положительные числа. Нестрого внутри этого прямоугольника отмечены K точек $p_1, p_2, \dots, p_K$ с целочисленными координатами. Точка p с целочисленными координатами, лежащая в A , называется **хорошей**, если расстояние от p до p_1 окажется не больше, чем расстояние от p до любой из точек $p_i, 1 \leq i \leq K$.

Внимание, вопрос: сколько существует хороших точек?

Формат входных данных

Первая строка входных данных содержит три целых положительных числа X, Y, K , $1 \leq X, Y, K \leq 2 \cdot 10^5$ — размеры прямоугольника и количество отмеченных точек. i -я из следующих K строк ($i = 1, 2, \dots, K$) содержит по два целых числа x_i, y_i ($0 \leq x_i \leq X, 0 \leq y_i \leq Y$) — координаты i -й точки. Гарантируется, что все точки попарно различны.

Формат выходных данных

Выведите одно целое неотрицательное число — ответ на задачу.

Примеры

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