

# Aware of depth

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Университет — это не только про учебу, но и про увлечения. Можно играть в футбол, заниматься киберспортом или опускаться в пещеры на сотни метров. Это увлечение называется спелеология, а люди, которые этим занимаются — спелеологи. То есть вы можете ~~забить на~~ совмещать учебу и заниматься лазанием в пещерах.

Группа из  $n$  студентов-новичков отправилась в экспедицию в пещеру со старшим инструктором для обучения спелеологии. Условия экспедиции следующие: необходимо опуститься в «колодец» (так называется вертикальный ход). Глубина «колодца» представляется целым числом  $d$  метров. Спуск в «колодец» происходит с помощью одной веревки или нескольких, которые можно связать между собой узлом. Всего у экспедитора есть  $m$  веревок, длина  $i$ -ой веревки измеряется целым числом метров  $l_i$ .

Так как группа состоит из новичков, инструктор решил, что использование тросов, состоящих из более чем двух веревок, опасно, поэтому для спуска в каждый из «колодцев» можно:

- использовать только **одну** веревку, длина которой в метрах не меньше глубины «колодца» — то есть для спуска в «колодец» можем использовать веревку  $j$ , если  $d \leq l_j$ ;
- или две** веревки, связанные между собой узлом, на который уходит по полметра от каждой веревки — то есть для спуска в «колодец» можем использовать веревки  $i$  и  $j$ , если  $d \leq l_j + l_k - 1$ .

При спуске каждый студент использует свой трос, состоящий из одной или двух веревок. Определите, существует ли способ спуститься в «колодец» всем  $n$  студентам, используя имеющиеся веревки. Учтите, что веревки нельзя резать, и выбранные для спуска веревки нельзя использовать повторно.

## Формат входных данных

В первой строке входных данных содержатся три целых числа  $n, m, d$  ( $1 \leq n, m \leq 10^5$ ) — количество студентов и количество веревок, а также ( $1 \leq d \leq 10^9$ ) — глубина «колодца». Во второй строке содержатся  $m$  целых чисел  $l$  ( $1 \leq l_i \leq 10^9$ ) — длины имеющихся веревок в метрах.

## Формат выходных данных

Выведите «YES», если все  $n$  студентов могут спуститься в «колодец», используя некоторые веревки, иначе выведите «NO».

## Примеры

стандартный ввод	стандартный вывод
2 3 4 4 2 2	NO
2 5 5 2 3 1 4 4	YES
5 1 10 1000000	NO
3 6 8 10 1 5 4 6 3	YES

## Замечание

Никогда заранее не знаешь, когда и где тебе пригодится знание алгоритмов.  
*Пока не спустишься в «колодец».*

# Bark Beetles

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Жуки-короеды являются серьезной проблемой для заповедников Томской области. Они залезают в кору деревьев и делают там свои дела нехорошие. Беспилотник пролетел над одним заповедником и сделал его снимки с высоты. Этот заповедник имеет форму прямоугольника и размер  $n \times m$  гектаров. Затем с помощью методов машинного обучения каждый гектар был промаркирован как лесная площадь или как болотистая площадь.

Известно, что некоторые гектары заражены жуком-короедом. Жук может совершать передвижения из одного гектара в другой, соседний с ним *по стороне*, причем, если некоторый гектар **леса** был подвержен обработке, то жук-короед **не может** переместиться в этот гектар, иначе жук может совершить перемещение, и посещенный гектар (леса или болота) также становится зараженным.

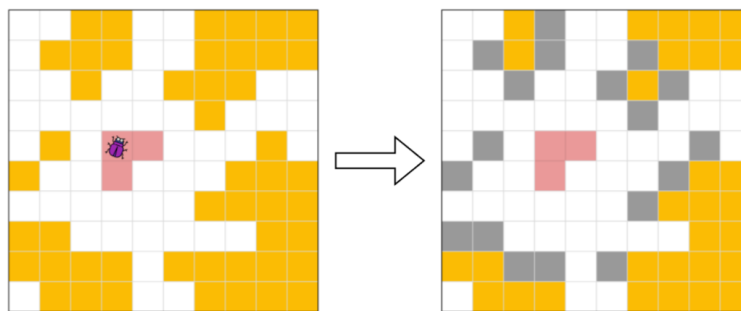
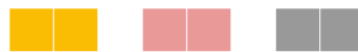


Иллюстрация к первому примеру. Гектары чистого леса показаны желтым цветом, изначально зараженные гектары леса показаны красным, остальное — болота. На изображении справа серым цветом покрашены лесные гектары, которые необходимо обработать к концу пятого дня.



При не цветной печати. Жёлтый, красный и серый цвета слева направо соответственно.

Требуется обезопасить все изначально незараженные гектары **лесной площади**, проведя на них профилактические работы (обработка болот не приносит результатов). Жук-короед не может распространяться моментально. За один день заражаются гектары леса, которые не были обработаны ранее, и хотя бы один соседний с ними гектар леса или болота был заражен в предыдущие дни, поэтому некоторые гектары леса можно обрабатывать с задержкой (не сразу за один день, но полностью до того, как до них доберется жук-короед).

Обработка совершается специальным дорогостоящим химикатом, поэтому вам необходимо снизить издержки. Вам необходимо определить, какое минимальное число изначально незараженных лесных гектаров необходимо обработать к концу каждого дня от 1 до  $n + m - 2$ , чтобы жук-короед не распространил заражение ни на один изначально незараженный лесной гектар. Считаем, что в начале каждого дня мы обрабатываем гектары, а затем в конце дня жук совершает распространение. Для большего понимания смотрите тестовые примеры.

## Формат входных данных

В первой строке входных данных содержатся два целых числа  $n, m$  ( $3 \leq n, m \leq 100$ ) — размеры в гектарах сторон прямоугольного заповедника.

Далее следуют  $n$  строк по  $m$  символов в каждой, описывающих начальное состояние заповедника в начале первого дня.

- Символом «x» обозначаются изначально зараженные гектары (неважно чего — лесов или болот);

- Символом «о» обозначаются изначально незараженные лесные гектары;
- Символом «.» обозначаются изначально незараженные гектары болот.

## Формат выходных данных

Выведите  $n + m - 2$  целых чисел,  $i$ -е число должно обозначать минимальное количество гектаров изначально незараженного леса, которое должно быть обработано к концу  $i$ -го дня для предотвращения распространения жука-короеда на каждом изначально чистом гектаре леса.

## Примеры

стандартный ввод	стандартный вывод
<pre> 10 10 ..oo..oooo .ooo..oooo .....ooo.. .....o... .o.xx...o. o..x...ooo .....oooo oo.....oo oooo.ooooo .ooo..oooo </pre>	<pre> 0 1 6 12 18 21 26 28 29 29 29 29 29 29 29 29 29 29 </pre>
<pre> 6 6 .xxooo ....oo ..... ooox.. oooo.. ...ooo </pre>	<pre> 3 3 6 9 9 9 9 9 9 9 </pre>
<pre> 4 16 .....xx x.....oo.....x xx.....oo..... .x..... </pre>	<pre> 0 0 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 4 </pre>

## Замечание

\*Вариант обработать химикатом изначально зараженные гектары, чтобы избавиться от проблемы на корню и предотвратить дальнейшие издержки, не рассматривается, так как это убьет очень ценный и уникальный для заповедника вид животных — томского жука-короеда.

\*Вывод в тестовых примерах сделан с переносом строки, чтобы он помещался в ячейки таблицы. Вы можете выводить числа в одну строку или в произвольное число строк — сравнение с ответом жюри осуществляется по токенам (переносы строки и пробелы не считаются токенами).

# Call him ВИТЁК

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт



Посвящение первокурсников ИШИТР.  
Фотозона. 29.11.2024

В нашей Инженерной Школе Информационных Технологий и Робототехники (ИШИТР) есть пять направлений, и каждое направление обозначается одной буквой в названии своей учебной группы. Так, например, студенты группы 8K03 обучаются на «Программной инженерии», а студенты группы 8B32 на «Информатике и вычислительной технике».

Каким образом эти буквы связаны с названиями направлений, является загадкой, но всего букв пять, как и направлений, и все вместе они образуют слово «ВИТЕК».

Ежегодно в середине первого семестра проходит «посвят» — или посвящение первокурсников. На очередном посвящении первокурсников каждое направление подготовило выступление, которое было оценено жюри из пяти человек. Оценкой выступления является целое число от 1 до 5. Каждый член жюри дал оценку выступлению каждого направления, причем все его оценки различны — т.е. каждый член жюри проранжировал выступления, где самое лучшее, по его мнению, получило оценку «5», а самое неудачное — оценку «1».

Однако под конец вечера в зал ворвался шестой член жюри! Он решительно идет к карточкам с цифрами, чтобы выставить свои оценки!

Вам известны баллы каждой команды до прихода шестого члена жюри. Определите, какое направление может забрать приз первого места в этот раз. Команда занимает первое место, если нет других команд, которые набрали строго больше баллов.

## Формат входных данных

Входные данные содержат пять строк по пять целых чисел в каждой — оценки, выставленные выступлению соответствующего направления. В первой строке содержатся оценки для направления «V», во второй — для «I», в третьей — для «T», в четвертой — для «E», и в пятой — для «K».

## Формат выходных данных

Выведите все буквы направлений, которые имеют шансы занять первое место, в порядке их появления в слове «ВИТЕК». Если вдруг все команды имеют шансы на первое место — выведите восклицательное «ВИТЕК!».

## Примеры

стандартный ввод	стандартный вывод
4 5 4 4 3 3 3 3 2 4 1 1 2 3 2 2 2 1 1 1 5 4 5 5 5	VK
2 5 1 3 4 3 1 2 4 5 5 2 4 1 2 4 3 3 2 3 1 4 5 5 1	VITEK!

# Dilemma

Имя входного файла:            стандартный ввод  
Имя выходного файла:        стандартный вывод  
Ограничение по времени:    0.5 секунд  
Ограничение по памяти:      16 мегабайт

**Обратите внимание на необычные ограничения по времени и памяти.**

В одном из учебных корпусов ТПУ студенты ИШИТР активно готовились к экзаменам. Чтобы систематизировать знания, они записывали ключевые понятия на листочках, обозначая каждую тему латинскими буквами (например, «a» — алгоритмы, «b» — базы данных и так далее).

Однако в процессе подготовки обнаружилась проблема: из-за дублирования тем и неоптимального ведения записей некоторые листочки стали избыточными. Чтобы упорядочить материал, студенты придумали алгоритм по удалению избыточных тем, работающий по следующим правилам:

- Разрешено удалять только по одной теме за операцию.
- Удаление допустимо исключительно при наличии в текущем наборе как минимум двух одинаковых тем (букв).
- Процесс продолжается до тех пор, пока возможны подобные удаления.

Помогите ребятам определить, какое минимальное количество тем им нужно будет выучить после применения всех допустимых операций в оптимальном порядке.

## Формат входных данных

Первая строка содержит целое число  $t$  — количество независимых наборов входных данных ( $1 \leq t \leq 100$ ). Каждая из следующих  $t$  строк содержит строку  $p$  ( $1 \leq |p| \leq 2 \cdot 10^5$ ) — начальный набор тем, для которых нужно посчитать ответ. Строка  $p$  состоит из строчных латинских букв.

Обращаем ваше внимание на то, что в этой задаче нет ограничения на суммарную длину всех строк в тестовом наборе.

## Формат выходных данных

Для каждого набора тем выведите одно число — минимальное количество тем, которое нужно будет выучить ребятам.

## Пример

стандартный ввод	стандартный вывод
3	5
abcde	7
dahjtof	1
aab	

# EcoHomes

Имя входного файла:  
Имя выходного файла:  
Ограничение по времени:  
Ограничение по памяти:

стандартный ввод  
стандартный вывод  
2 секунды  
256 мегабайт

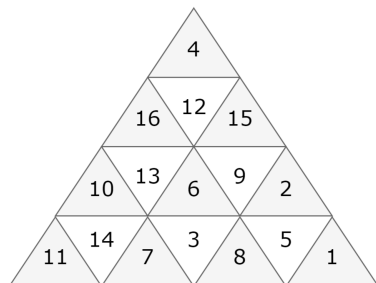


Иллюстрация ко второму примеру.

У вас дома нашлось  $n^2$  хомяков (случайно), и вы решили поселить их в EcoHomes — систему клеток-домиков для домашних животных, где каждая клетка в профиль выглядит как равнобедренный треугольник (а в пространстве такая клетка выглядит как призма). И каждая такая клетка станет домиком только для одного хомячка.

Одинаковый размер клеток позволяет создавать многоуровневые конструкции: если у вас есть  $n^2$  клеток, вы можете создать большой треугольник клеток, состоящий из  $n$  уровней! Это очень хорошая идея, так как такая конструкция экономит место в доме.

Единственная неприятность — хомяки в соседних по стороне клетках могут не дружить. Известно, что два хомяка с номерами  $i$  и  $j$  конфликтуют, если *не выполняется* следующее условие:  $3 \leq |i - j| \leq \frac{n^2}{2}$ , при этом хомяки  $i$  и  $j$  находятся в соседних по стороне клетках; иначе хомячки имеют same vibe и живут мирно.

Ваша задача — определить расположения хомяков в клетках так, чтобы каждая пара хомяков в соседних по стороне клетках не конфликтовала. Если возможных расположений несколько — подойдет любое, а если ни одного такого расположения не существует, то об этом тоже нужно сообщить.

## Формат входных данных

В единственной строке входных данных содержится единственное целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество уровней в большом треугольнике, который вы составили из одиночных клеток для хомяков.

## Формат выходных данных

Если не существует ни одного расположения, при котором не будет конфликтов между хомяками в соседних по стороне клетках, выведите «-1». Иначе выведите  $n$  строк, в  $i$ -й строке должно быть  $2i - 1$  целых чисел — номера хомяков в соответствующих клетках  $i$ -го уровня слева направо. Для лучшего понимания рассмотрите вывод во втором тестовом примере и иллюстрацию выше.

## Примеры

стандартный ввод	стандартный вывод
1	1
4	4 16 12 15 10 13 6 9 2 11 14 7 3 8 5 1
2	-1

## Замечание



Чтобы легенда была связана с университетом, предположим, что это особые кибер-хомяки, необходимые для нужд ИШИТР.

# Finding Technodromes

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Студенты ИШИТР, создавая роботов, дают им особые имена. Имя считается палиндромом, если оно одинаково читается как слева направо, так и справа налево (как в примерах «level», «rotor», «x»).

Часть имени робота (подслово) называется «технодромом» для числа  $n$ , если в ней можно изменить не более  $n$  символов, чтобы получить палиндром. Например, при  $n = 2$  подслова «motor», «circuit», «rotor» являются «технодромами».

Подслово можно получить, удалив символы (включая ноль) с начала и/или конца исходного имени. Например, для имени «tpru» подсловами будут «t», «p», «u», «tp», «pu», «tpru».

Необходимо определить количество подслов имени робота  $s$ , которые являются «технодромами» для заданного числа  $n$ .

## Формат входных данных

В первой строке входных данных содержатся два целых числа  $k, n$  — длина слова ( $1 \leq k \leq 5000$ ) и значение  $n$  ( $0 \leq n \leq k$ ). Во второй строке содержится слово  $s$ , состоящее из  $k$  строчных латинских букв.

## Формат выходных данных

Количество подслов имени робота  $s$ , которые являются «технодромами» для числа  $n$ .

## Примеры

стандартный ввод	стандартный вывод
3 0 aba	4
5 1 aabba	14
5 2 edcba	15

## Group guide

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Ни для кого не секрет, что в Томском Политехе учиться сложно. Каждую неделю студенты погружаются в решение сложнейших задач и написание проектов, но больше всего — в лабораторные работы.

Однажды в одной из групп студенты собрались в уютной кофейне, чтобы обсудить, как им справиться с накопившимися задачами. За столом сидели Слава, Костя и Влад.

Влад, давно мечтавший о карьере инженера, с воодушевлением стал рассказывать о своих идеях. Он предложил ребятам разбиться на мини-группы для выполнения лабораторных работ, ведь работа в команде всегда придает сил и вдохновения. Ребята поддержали идею друга, но чтобы сделать это рационально, они решили соблюсти два условия:

- Все  $n$  студентов группы должны разбиться на мини-группы по 2 или 3 человека. Каждый студент принадлежит ровно одной мини-группе.
- Каждая мини-группа берет выполнять одну из  $m$  лабораторных работ.
- Никакие две мини-группы не должны выполнять одну и ту же лабораторную работу. Некоторые работы могут остаться невыбранными.

Друзья очень любят решать сложные задачи, поэтому им стало интересно, сколько существует способов разбить  $n$  студентов на мини-группы и распределить между ними  $m$  работ, соблюдая указанные условия.

Помогите ребятам решить данную задачу!

### Формат входных данных

В первой строке входных данных содержатся два числа — количество студентов  $n$  ( $3 \leq n \leq 100$ ) и количество лабораторных работ  $m$  ( $1 \leq m \leq 100$ ).

### Формат выходных данных

Выведите единственное целое число — ответ на задачу по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
3 3	3
6 6	2100
5 2	20
4 1	0

### Замечание

Рассмотрим третий пример. Обозначим студентов числами от 1 до 5. Тогда есть 10 разбиений студентов на мини-группы:

$([1,2,3], [4,5]), ([1,2,4], [3,5]), ([1,2,5], [3,4]),$   
 $([1,3,4], [2,5]), ([1,3,5], [2,4]), ([1,4,5], [2,3]),$   
 $([2,3,4], [1,5]), ([2,3,5], [1,4]), ([2,4,5], [1,3]),$   
 $([3,4,5], [1,2]).$

Внутри каждого разбиения есть 2 варианта распределения работ. Следовательно, получается 20 вариантов.



# Hack this shift

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.05 секунд  
Ограничение по памяти: 52 мегабайта

Как вы все знаете и понимаете, скоро нас всех заменит искусственный интеллект, и мы останемся без работы (нет). Преподаватели будут составлять задания с помощью нейросетей, а студенты будут с помощью нейросетей их выполнять (или уже?). Нужно еще немного посидеть тихо, и тогда заживем. Вот лично мне уже сложно конкурировать с GPT2 от OpenAI, а это ведь только начало...

Тем не менее многие вещи остаются все еще недоступными для ИИ: например, написание симфоний, сложные когнитивные задачи, нестандартные головоломки и что произошло на какой-то там площади.

Ваша задача — понять правило преобразования входных строк в выходные на основе тестового примера и написать алгоритм, который реализует это правило. Вот так вот!

## Формат входных данных

В первой строке входных данных содержится целое число  $t$  ( $1 \leq t \leq 10$ ) — количество наборов входных данных. Далее следуют  $t$  строк, в каждой из которых содержится строка, состоящая из букв  $[a - e]$ , длиной от 1 до 5 символов.

## Формат выходных данных

Выведите  $t$  строк, в каждой  $i$ -й строке выведите преобразованную  $i$ -ю строку из входных данных. Гарантируется, что значение  $t$  не используется в правиле преобразования, каждая строка подвергается обработке независимо.

## Пример

стандартный ввод	стандартный вывод
9	caada
bdcea	aadab
edabb	babea
addaa	caceb
bdeab	caed
bdbe	aead
ecce	bb
ae	bb
ae	bb
ae	

# Interactive measure

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

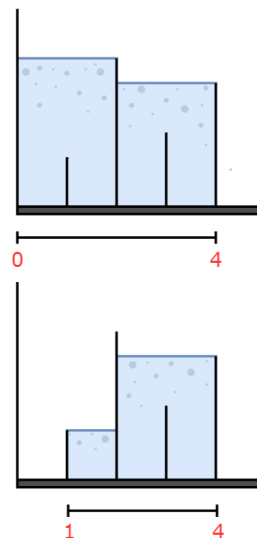
Это интерактивная задача.

У нас есть специальный аквариум для рыбок. Но мы вам его не покажем. Он похож на обычный аквариум: сделан в форме коробки без верхней грани. Длина аквариума —  $n$ , ширина — 1, а высота — 100. Аквариум разделен на  $n - 1$  частей  $n$  вертикальными перегородками. Расстояние между каждой парой соседних перегородок — 1. Левая стенка аквариума является перегородкой номер 0, а самая правая — перегородкой номер  $n - 1$ .

Все перегородки между номерами 0 и  $n - 1$  могут иметь разную целочисленную высоту от 1 до 99.

Вам нужно определить высоту всех перегородок не более чем за  $n$  запросов. За один запрос можно выбрать пару перегородок  $l < r$  и узнать, какой максимальный объем может занимать вода, если наливать ее между перегородками  $l$  и  $r$  так, чтобы вода не переливалась через перегородки  $l$  и  $r$ .

Гарантируется, что все перегородки имеют целочисленную высоту, что перегородок хотя бы 3, что перегородки номер 0 и номер  $n - 1$  имеют высоту 100, что все остальные перегородки строго ниже их, но минимум высоты 1.



Пример запросов и физики воды.

## Формат входных данных

В единственной строке входных данных содержится целое число  $n$  ( $3 \leq n \leq 1000$ ) — число перегородок в аквариуме. Гарантируется, что всегда существует способ узнать высоты всех перегородок не более чем за  $n$  запросов.

## Протокол взаимодействия

Жюри имеет целое число  $n$  и список высот перегородок  $h$  из  $n$  элементов, которые постоянны для одного теста. Далее жюри передает число  $n$  в поток ввода для вашей программы, список  $h$  от вас скрыт — вы должны его узнать. Интерактор в этой задаче **не адаптивен** — список  $h$  не меняется в процессе взаимодействия.

После этого вы можете выбрать два целых числа  $l, r$  и вывести следующее на отдельной строке, чтобы задать запрос:

- «?  $l$   $r$ » ( $0 \leq l < r < n$ ): узнать максимальный объем воды, помещающийся между перегородками  $l$  и  $r$ .

Затем интерактор отвечает следующим образом:

- Если ваш запрос некорректный, или вы превысили число запросов, то взаимодействие прерывается без обратной связи, и вы получаете вердикт «WA» за задачу;
- Если запрос корректный, интерактор отвечает целым числом, обозначающим максимальный объем воды, который помещается между перегородками  $l$  и  $r$ .

Если вы хотите напечатать ответ, вы можете вывести следующее на отдельной строке:

- «!  $h_0$   $h_1$  ...  $h_{n-1}$ » ( $h_0 = h_{n-1} = 100$ ;  $1 \leq h_i < 100$  при  $1 \leq i < n - 1$ ): где  $h_i$  является высотой перегородки  $i$ .

Затем происходит следующее взаимодействие:

- Если ваш ответ некорректный, то взаимодействие прерывается без обратной связи, и вы получаете вердикт «WA» за задачу;
- Если ответ правильный, то взаимодействие прерывается без обратной связи, и вы получаете вердикт «ОК» за тест. Вердикт «ОК» за задачу выставляется только в случае получения вердиктов «ОК» на всех тестах.

Обратите внимание, что печать ответа **не** учитывается в количестве сделанных запросов.

После вывода каждого запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт Решение «зависло». Для этого используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `sys.stdout.flush()` в Python;
- смотрите документацию для других языков.

## Пример

стандартный ввод	стандартный вывод
5	? 1 2
2	? 2 3
3	? 0 2
8	! 100 2 4 3 100
ok	

## Замечание

Лишние переводы строк в примере показаны для большей наглядности процесса взаимодействия. Интерактор «делает первый ход», передавая в поток ввода вашей программы число  $n$ , далее ваша программа делает запрос, затем «ход переходит» к интерактору и он отвечает на запрос и т.д.

Не забывайте, что существует пробный тур, доступный на всем протяжении основного тура, в котором есть интерактивная задача с примером кода.

Чтобы легенда была связана с университетом, предположим, что это особый кибер-аквариум (с кибер-рыбами), необходимый для нужд ИШИТР.

# Jackbox at the party!

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

Часто в ИШИТР проходят студенческие вечеринки, и обычно они приурочены к какому-нибудь особому дню: например, ко дню программиста (256-й день года) или ко дню рождения АВТФ (факультет автоматике и вычислительной техники). И обычно на таких вечеринках студенты любят играть в `Jackbox`.

И вот в одном раунде проигравший игрок попал в темницу, выбраться из которой он может только с помощью своей удачи: у игрока есть  $n$  артефактов, которые он собрал за всю игру, каждый имеет свою ценность  $a_i$ , выражающуюся натуральным числом. Страж игры случайным образом выбирает два артефакта  $i$  и  $j$  из  $n$  артефактов игрока, затем перемножает их ценности, получая число  $s = a_i \times a_j$ . И если  $s$  делится нацело на число игроков не в темнице, равное  $k$ , то игрок освобождается из темницы и продолжает игру, но лишается всех артефактов, а ценность  $s$  распределяется поровну между  $k$  оставшимися игроками.

Определите вероятность, с которой игрок выйдет из темницы.

## Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $2 \leq n \leq 200000$ ) — количество артефактов игрока, заключенного в темнице. Во второй строке входных данных содержатся  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — ценность каждого артефакта. В третьей строке содержится целое число  $k$  ( $1 \leq k \leq 10^5$ ) — число игроков вне темницы.

## Формат выходных данных

Выведите вероятность с точкой в качестве разделителя.

Ваш ответ будет засчитан, если относительная или абсолютная погрешность не будет превышать  $10^{-9}$ . Формально, если  $a$  — ваш ответ, а  $b$  — ответ жюри, то он будет засчитан, если  $\frac{|a-b|}{\max(b, 1)} \leq 10^{-9}$ .

## Примеры

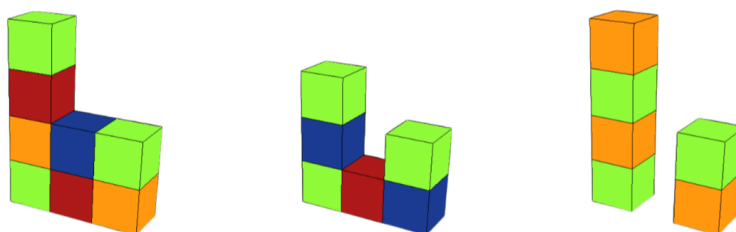
стандартный ввод	стандартный вывод
3 2 4 3 6	0.66666666666667
4 3 5 2 4 10	0.33333333333333
5 12 1 10 7 6 13	0.000000000000000000000001

# Kubiki for TikTok House

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

TikTok-хаус AVTFamily очень устал, так как сегодня они сняли уже два ролика. Ребята решили отдохнуть и поиграть с кубиками. Есть разноцветные кубики  $m$  различных цветов. Кубиков цвета  $i$  ровно  $c_i$  штук. Кубики можно ставить друг на друга или в слоты на поверхности. Всего на поверхности есть три слота, расположенные в ряд вплотную друг к другу. *Постройкой* назовем любую конструкцию **из всех** имеющихся кубов.

Недопустимо, чтобы соседние по стороне или ребру кубики имели одинаковый цвет. Определите, может ли тогда существовать хотя бы одна *постройка*, удовлетворяющая этому условию. Как только будет обнаружена хотя бы одна такая *постройка*, будет снят соответствующий вирусный рилс на 50 лайков, иначе ребята удаляют страничку и прилагают больше усилий к учебе.



Примеры башен для первых трех тестовых примеров. Обратите внимание, что в третьей *постройке*, так же как и в первых двух, задействованы три слота, но средний слот оставлен пустым.

## Формат входных данных

В первой строке входных данных содержится целое число  $m$  ( $1 \leq m \leq 2 \times 10^5$ ) — количество различных цветов. Во второй строке входных данных содержатся  $m$  целых чисел  $c$  ( $1 \leq c_i \leq 10^9$ ) — количество кубиков каждого из  $m$  цветов.

## Формат выходных данных

Выведите единственную строку «YES», если существует хотя бы одна *постройка*, удовлетворяющая условию задачи; иначе выведите «NO».

## Примеры

стандартный ввод	стандартный вывод
4 1 3 2 2	YES
3 3 2 1	YES
2 3 3	YES
3 1 52 8	NO

# Lunar debate

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

На скучной паре два студента Костя и Маша увлеклись странной игрой в тетрадке. Они рисовали точки, представляя их как колонии на далеких планетах, а между ними строили гипертоннели для межгалактических перелётов. По правилам их игры гипертоннель можно проложить только между двумя колониями, если расстояние между ними *не больше*  $h$  световых лет. Но здесь был подвох: чем больше  $h$ , тем дороже строительство тоннеля.

Студенты поспорили, кто быстрее найдёт минимальное значение  $h$ , при котором все колонии окажутся связанными в единую сеть, иначе говоря: чтобы из любой колонии можно было добраться до любой другой через цепочку тоннелей. Чтобы разрешить спор, они решили написать программу, которая по координатам колоний точно вычислит минимальное значение  $h$ . Помогите ребятам написать программу!

На плоскости задано  $n$  точек, каждая из которых имеет свои координаты  $(x_i, y_i)$ . Гипертоннели можно строить только между парами колоний, расстояние между которыми *не больше*  $h$  световых лет. Найдите минимальное значение  $h$ , при котором все колонии становятся связными (*любые две колонии соединены цепочкой тоннелей*).

## Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $2 \leq n \leq 1000$ ) — количество точек на плоскости. В следующих  $n$  строках содержатся пары чисел  $(x_i, y_i)$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ) — координаты точек. Гарантируется, что все пары точек во входных данных различны.

## Формат выходных данных

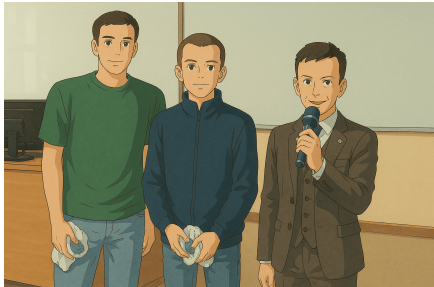
Необходимо вывести одно число — минимальное значение  $h$ , с точностью не менее 6 знаков после запятой.

## Примеры

стандартный ввод	стандартный вывод
4 0 0 0 1 1 0 1 1	1.000000000000
4 1 2 3 4 5 6 1 9	5.000000000000

# MSIT MStIT

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт



Конференция MSIT. День закрытия. 18.04.2024

Каждый год весной ИШИТР ТПУ проводит конференцию «Молодежь и современные информационные технологии» (сокращенно «МСИТ»).

На ней студенты, аспиранты и молодые ученые представляют свои научные или учебные работы всем заинтересованным и обмениваются опытом с коллегами.

Однако это очень сложное мероприятие для принимающей стороны, в частности, в подготовке места проведения. Поэтому в этой задаче вы будете регулировать высоту скамеек в аудитории.

В этом году на конференции собралось очень большое количество докладчиков. И, чтобы уместить их всех в одном месте, была занята самая большая лекционная аудитория. Аудитория состоит из  $n$  рядов, расположенных друг за другом, на каждом ряду есть длинная скамейка, на которой могут располагаться слушатели.

Сейчас ряд  $i$  имеет высоту  $h_i$  относительно пола. Для комфорта слушателей нужно сделать  $h_i \leq h_{i+1}$  для всех  $1 \leq i < n$ . Можно потратить  $c_i$  денег из бюджета, чтобы уменьшить или увеличить высоту  $h_i$  на 1 для любого  $1 \leq i \leq n$ . Операции по изменению высоты можно проводить любое число раз с любыми рядами. Например, если вы хотите изменить высоту ряда  $i = 1$  с  $c_1 = 3$  на 2 единицы и высоту ряда  $j = 2$  с  $c_2 = 5$  на 3 единицы, то вы должны заплатить 21.

Определите минимальное число денег для преобразования рядов согласно условию задачи (чтобы выполнялось  $h_i \leq h_{i+1}$  для всех  $1 \leq i < n$ ).

## Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $1 \leq n \leq 1000$ ). Во второй строке входных данных содержатся  $n$  целых чисел  $h$  ( $1 \leq h_i \leq 10^9$ ) — начальная высота рядов. В третьей строке содержатся  $n$  целых чисел  $c$  ( $1 \leq c_i \leq 10^3$ ) — стоимость изменения высоты  $i$ -го ряда на единицу высоты.

## Формат выходных данных

Выведите одно целое число — минимальное число денег для преобразования высоты рядов.

## Примеры

стандартный ввод	стандартный вывод
4 1 3 2 3 1 1 1 1	1
6 1 3 2 3 5 1 1 4 2 1 2 3	12
3 1 3 2 2 4 8	4