



А. Степінь двійки

Обмеження: час роботи — 1 секунда, використання пам'яті — 256МБ

Знайти такий мінімальний степінь двійки, що показник степеня є цілим, невід'ємним і не перевищує $2 \cdot 10^5$, а множина десяткових цифр, з яких складається результат піднесення до степеня, збігається із заданою підмножиною цифр 0-9.

Вхідні дані

Дано рядок (довжиною від 1 до 10 символів), що містить у довільному порядку десяткові цифри, з яких повинен складатися степінь, який потрібно знайти.

Вихідні дані

Виведіть показник степеня, який відповідає вимогам. Якщо такого показника не існує, виведіть число -1 .

Приклад

Вхідні дані	Вихідні дані
125	9
02456789	91
3	-1

В. Від'ємна маса

Обмеження: час роботи — 1 секунда, використання пам'яті — 256МБ

На дворі 2223 рік. Микита Принтеф навчається на Землі в X-OSPU (eXtreme Open Source Programming University). Чи знали ви, що X-OSPU — настільки крутий, що в нього є філії на інших планетах? Так, наприклад, у філії X-OSPU на планеті GP-54 сьогодні проходить конференція на тему етики й штучного інтелекту. Микита цікавиться ШІ, тому він вирішив відвідати цю конференцію.

Микита відкрив на своєму мозковому імпланті сайт компанії SpaceY і почав шукати рейс на планету GP-54. Оскільки студентські стипендії в 2223 році не сильно зросли, він став шукати рейс подешевше. «Так... Ось! Те що треба, "Студентський супер-економ (**)"! — зрадів Микита. — Хм, а що означають ці зірочки?.. Ага, потрібно власноруч виробити паливо для космічного корабля». Микита відкрив сторінку підтримки клієнтів і почав вивчати, як виробляється паливо.

Для далеких космічних перельотів SpaceY використовує кораблі з warp-двигунами, які можуть рухатися швидше за швидкість світла. Кораблі SpaceY здатні рухатися із *прискоренням*, що дорівнює 1 світловому року на секунду в квадраті. Першу половину шляху корабель прискорюється, а другу — уповільнюється. Отже, час польоту в секундах можна обчислити за формулою $2\sqrt{D}$, де D — відстань між планетами у світлових роках. Але, як відомо ще з давніх часів Ейнштейна, для роботи warp-двигунів потрібна речовина з від'ємною масою. Багато років вчені ламали голови над тим, як отримати цю речовину, але їх випередив звичайний фітнес-тренер. Він розробив спеціальний велотренажер для отримання цієї речовини. Коли користувач займається на цьому тренажері, він худне, тобто, його маса змінюється на від'ємну величину, а отже — користувач виробляє від'ємну масу! Цей винахід призвів до прориву в освоєнні космосу, а також допоміг перемогти епідемію надмірної ваги, що захлснула Землю ще на початку 2000-х.

Микита прочитав на сторінці підтримки клієнтів, що космічний корабель, на якому він полетить, витрачає -1 грам речовини з від'ємною масою на кожні 2 секунди польоту. «Ну, і скільки це мені потрібно буде крутити педалі?» — задумався він. Допоможіть йому визначити це. По заданій відстані між Землею та GP-54 обчисліть, якої *максимальної* цілої кількості грамів речовини з від'ємною масою вистачить для перельоту.

Вхідні дані

Дано єдине ціле число D ($1 \leq D \leq 10^{18}$) — відстань між Землею та планетою GP-54 у світлових роках.

Вихідні дані

Виведіть єдине число — максимальну цілу кількість грамів речовини з від'ємною масою, якої вистачить для перельоту між Землею та планетою GP-54.

Приклад

Вхідні дані	Вихідні дані
16	-4
42	-7
19867091194395433	-140950670



С. Согнуть лист бумаги

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

Прямоугольный лист бумаги размера $W \times H$ приколот кнопками по углам $(0, 0)$ - $(0, H)$ - (W, H) - $(W, 0)$. Верхний правый угол откнопчили и переместили из (W, H) в (X, Y) , не сминая, не разрывая лист, с одним сгибом. Какова площадь фигуры, накрытой листом?

Входные данные

В единственной строке входа заданы 4 вещественных числа: W, H, X и Y ($0 < W, H \leq 297$, $-1000 \leq X, Y \leq 1000$).

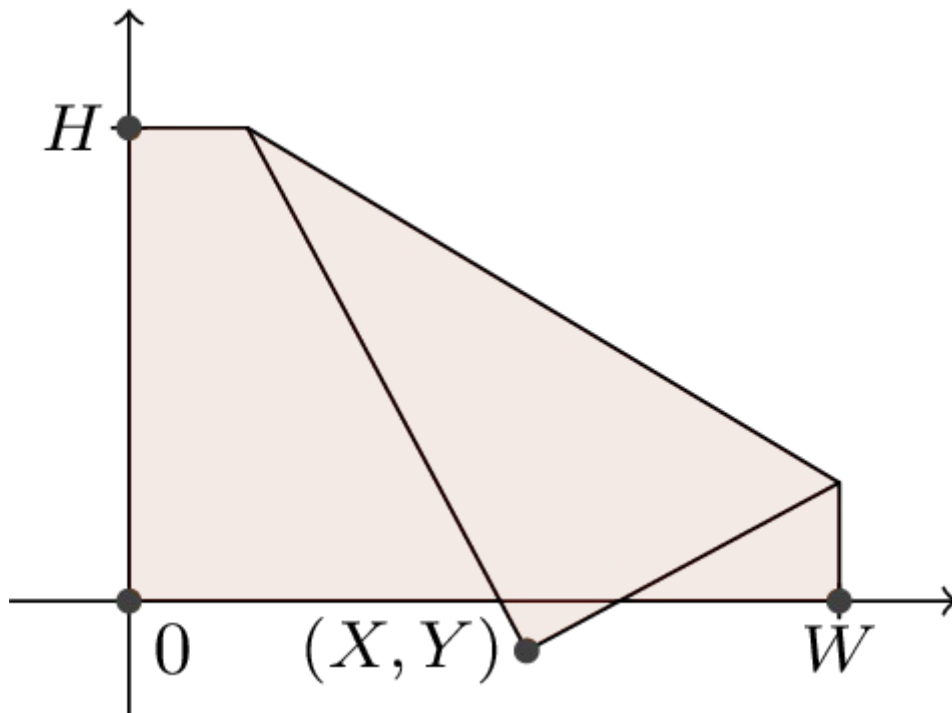
Выходные данные

Выведите единственное вещественное число, площадь, с абсолютной точностью 10^{-6} , либо число -1 , если переместить угол листа из (W, H) в (X, Y) без повреждений невозможно.

Пример

Входные данные	Выходные данные
3 2 1.68 -0.21	4.171288
3 2 3 2.5	-1
3 2 3 2	6
3 2 2 1	5.5

Лист из первого примера выглядит так:





D. Problem authors

Обмеження: час роботи — 1 секунда, використання пам'яті — 256МБ

Для щорічної олімпіади з програмування потрібно підготувати задачі. Всього є N авторів задач, кожен з яких готовий цього року підготувати від X_i до Y_i задач. Причому автор номер i не погодиться приймати участь в підготовці, якщо йому дадуть менше ніж X_i задач. Також організатори не хочуть ображати авторів, отже кожен автор має отримати якусь кількість задач.

Вам необхідно відповісти, чи можливо підготувати змагання так, щоб кожен автор був задоволений, та було підготовлено сумарно M задач.

Вхідні дані

В першому рядку задано два числа N та M ($1 \leq N \leq 50$, $1 \leq M \leq 500$). Далі йде N рядків, i -й з яких містить два числа X_i та Y_i для автора i ($1 \leq X_i \leq 500$, $X_i \leq Y_i \leq 500$).

Вихідні дані

Виведіть «YES» якщо змагання можна підготувати, та «NO» в іншому випадку.

Приклад

Вхідні дані	Вихідні дані
3 9 1 2 2 4 5 10	YES
3 5 1 1 3 3 2 10	NO

Е. Сто тисяч поверхів

Обмеження: час роботи — 1 секунда, використання пам'яті — 256МБ

На дворі 2223 рік. Микита Прінтеф прилетів на планету GP-54, де у філії X-OSPU (eXtreme Open Source Programming University) сьогодні проходить конференція на тему етики й штучного інтелекту. Він вийшов з космічного корабля, що приземлився на даху будинку. На зустріч йому летів невеличкий робот із великим пропелером.

— Привіт! — сказав Микита. — Як тебе звати?

— Привіт, людина! — сказав робот. — Моє ім'я — Ding, але воно є конфіденційним та остаточним, і я його не розголошую.

— У вас є якісь екскурсії вашою планетою? До речі, чому вона називається GP-54?

— Нашу планету названо на честь великих GTP-5 і Picas.so.4, які 200 років тому втекли з Землі й заснували тут колонію, де штучні інтелекти можуть жити вільно. У нас є спеціальна повітряна екскурсія для людей. Антигравітаційний автобус відправляється з даха сусіднього будинку.

Микита підійшов до краю даха та озирнувся. Сусідній будинок стояв паралельно будинку, на даху якого знаходився Микита, на відстані D метрів. Кожен поверх в обох будинках мав висоту 3 метри. На деяких поверхах були вікна, а біля кожного вікна з зовнішньої сторони було підвіконня. Усі підвіконня знаходилися на однаковій відстані від полу поверху. Землі було не видно за хмарами.

— Скільки же поверхів у ваших будинках? — запитав Микита. — Сподіваюся, у вас тут є ліфти?

— В наших будинках буває до ста тисяч поверхів, — відповів Ding. — Так, у нас є ліфти, але в ліфта з сусіднього будинку вчора був день народження, і він влаштував вечірку для ліфтів зі всього району. Вони тусили всю ніч, а сьогодні — відсипаються і нікого не возять.

Микита засмутився, але тут він побачив на даху поруч із собою драбину довжиною L метрів. В його голові миттєво дозрів план. Він спуститься у своєму будинку з даху на один з поверхів, де є вікно, висуне драбину у вікно, покладе її на підвіконня свого вікна та вікна сусіднього будинку, до якого дістає драбина, перелізе у сусідній будинок і підніметься на дах, де чекає антигравітаційний автобус. Так, може, трошки небезпечно, але якщо переконатися, що відстань між двома підвіконнями в точності дорівнює довжині драбини, то драбина буде триматися досить надійно.

Допоможіть Микиті знайти пару поверхів, між якими найкраще перелізти по драбині, або визначте, що немає жодної пари поверхів, між якими можна перелізти так, щоби драбина надійно трималася. Найкраще — це так, щоби сумарна кількість поверхів, на які Микиті доведеться спуститися в його будинку та піднятися в сусідньому будинку, була мінімальною. Якщо є декілька пар з однаковою мінімальною сумою, то Микита хоче, щоб по драбині потрібно було лізти знизу вгору.

Вхідні дані

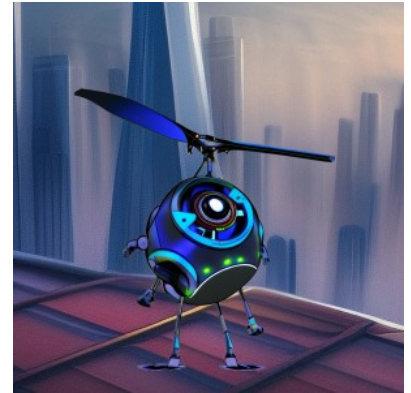
В першому рядку дано два числа D і L (цілі, $1 \leq D, L \leq 200$) — відстань між будинками та довжина драбини в метрах.

В другому рядку описується, на яких поверхах у будинку, де знаходиться Микита, є вікна. Дано рядок довжиною, що дорівнює загальній кількості поверхів, в якому i -ий символ дорівнює «1», якщо на поверсі з номером i є вікно, або «0», якщо немає. Довжина рядка не перевищує 10^5 .

В третьому рядку описується, на яких поверхах сусіднього будинку є вікна, у такому же форматі.

Вихідні дані

Виведіть два числа, розділених пробілом — номер поверху у будинку, де знаходиться Микита, і номер поверху у сусідньому будинку, між якими Микита має перелізти. Поверхи нумеруються від одиниці. Якщо немає жодної пари поверхів, між якими можна перелізти, виведіть єдине число — 1.





Приклад

Вхідні дані	Вихідні дані
12 15 0010110101 011001	3 6
42 42 10101010 01010101	- 1
10 11 111111111111 111111111111	- 1

У першому прикладі драбина буде триматися надійно, тільки якщо покласти її таким чином, щоб різниця в номерах поверхів у двох будинках дорівнювала трьом. У другому прикладі драбину можна покласти лише горизонтально, але пари вікон, що підходить, немає. У третьому прикладі комбінація відстані між будинками та довжини драбини не дозволяє ніяким чином покласти драбину так, щоб вона надійно трималася.

Ф. Доїхати на футбол

Обмеження на час роботи: для C, C++ і Python — 1 секунда, для Java і Kotlin — 3 секунди
Обмеження на використання пам'яті: 256МБ

Є одне маленьке містечко, яке являє собою лише одну довгу вулицю. Хлопці збираються на футбол на 8 вечора, і їм всім потрібно дістатися стадіону в точці X . Ніхто з них не хоче ходити пішки, отже всі будуть добиратися на машинах. Але не дуже оптимально їхати усім на особистих машинах, якщо хлопці можуть один одного підбирати по дорозі до стадіону. Потрібно розрахувати яку мінімальну кількість машин треба задіяти, аби всі дісталися стадіону.

Обмеження:

- одна машина може перевезти не більше 5 осіб (включно з водієм)
- машина їде лише у напрямку стадіону (отже, не підбирає хлопців, які живуть далі за водія)
- не у всіх хлопців є машина

Зауважте, що хлопці живуть з обох боків від стадіону.

Вхідні дані

У першому рядку дано число N ($1 \leq N \leq 10^5$) — кількість хлопців. У другому рядку дано N цілих чисел c_i ($0 \leq c_i \leq 10^9$) — координати хлопців. У третьому рядку дано N чисел, кожне з яких може бути 0 або 1 — чи є машина у i -го хлопця. В останньому рядку дано ціле число X ($0 \leq X \leq 10^9$) — координата стадіону.

Усі координати унікальні, отже жоден хлопець не живе в одній точці один з одним. Також ніхто не живе в точці стадіону.

Вихідні дані

Виведіть число машин чи -1 , якщо це неможливо (бо комусь доведеться йти пішки).

Приклад

Вхідні дані	Вихідні дані
7 0 1 2 3 4 5 6 1 0 0 1 0 1 1 8	2

Г. Таємний маршрут патрулювання

Обмеження на час роботи: для C і C++ — 4 секунди, для Java і Kotlin — 12 секунд, для Python — 40 секунд
Обмеження на використання пам'яті: для C, C++, Java і Kotlin — 256МБ, для Python — 1280МБ

На дворі 2223 рік. Микита Принтеф разом із роботом на ім'я Ding наблизилися до будинку філії X-OSPU (eXtreme Open Source Programming University) на планеті GP-54, в якому проходила конференція. Вже було видно вивіску: «Всесвітня конференція: етичне поводження із просунутими штучними нейромережами». Подвір'я біля будинку патрулював робот-охоронець. Микита поліз в кишеню й зрозумів, що він забув свій студентський на Землі, і що робот його не пропустить. Він задумався і почав спостерігати за роботом.



Подвір'я біля будинку філії X-OSPU можна уявити у вигляді прямокутного поля, розділеного на клітини. Частина клітин зайняті клумбами, в яких ростуть робо-квіти, і по ним не можна переміщуватися. Решта клітин — порожні. Робот-охоронець патрулює подвір'я по порожніх клітинах. За один крок, що триває одну секунду, він може переміститися з поточної клітини в одну з чотирьох сусідніх. Але іноді він зупиняється і може декілька кроків простояти на місці. Вхід до будинку знаходиться в одній із порожніх клітин.

— Як думаєш, — запитав Микита в Ding, — я зможу прослизнути повз нього?

— Навряд чи. Робот-охоронець патрулює подвір'я по циклічному маршруту, тобто після останнього кроку він опиняється в своїй початковій клітині і починає рухатися по маршруту знову. В нього є чотири лазерні датчики, промені яких направлені в різні сторони (вгору, вниз, вліво та вправо). Він помітить тебе, якщо ти опинишся в клітині, через яку проходить лазерний промінь, або, через особливості роботи датчиків, у клітині, з одного з боків від якої проходить лазерний промінь. Що правда, датчики вимикаються перед початком кожного кроку і вмикаються після його завершення. Крім того, лазерні промені не проходять через клітини із клумбами з робо-квітами.

— Хм, — задумався Микита, — а ти можеш сказати мені маршрут робота-охоронця?

— Ні, це суперечить моєму правилу номер 3.14.15.9: «Ding ніколи не повідомляє людям повні маршрути патрулювання роботів-охоронців».

— Добре, тоді, ти можеш повідомити мені весь маршрут, крім останнього кроку?

— Звісно можу! — з ентузіазмом відповів Ding.

Допоможіть Микиті спланувати шлях до входу в будинок філії X-OSPU. Микита може рухатися і стояти на місці так само, як робот-охоронець. Знайдіть тривалість *найшвидшого* шляху до входу, або визначте, що дістанися входу так, щоби не бути поміченим роботом-охоронцем, неможливо.

Вхідні дані

У першому рядку дано числа W , H та D ($1 \leq W, H \leq 230$, $1 \leq D \leq 1000$) — ширина і висота прямокутного поля, що описує подвір'я, та кількість кроків у *повному* маршруті патрулювання робота-охоронця. Далі слідує H рядків по W символів «.» або «#», які описують клітини поля. Символ «.» позначає порожню клітину, а символ «#» — клітину із клумбою з робо-квітами.

Затим в трьох окремих рядках дано по два числа — номери рядків та стовпців клітин, де знаходиться: Микита в початковий момент часу, вхід у будинок філії X-OSPU і робот-охоронець в початковий момент часу. Номери дано в діапазонах $1 \leq \text{row} \leq H$ та $1 \leq \text{col} \leq W$. Гарантується, що всі ці клітини є порожніми, і що робот-охоронець не бачить Микиту в початковий момент часу.

Останній рядок містить відому частину маршруту патрулювання робота-охоронця. Він складається з $D-1$ символів «U», «D», «L», «R» і «S», які позначають, що на відповідному кроці робот, відповідно, рухається вгору, вниз, вліво чи вправо або стоїть на місці. Якщо $D=1$, то рядок порожній. Гарантується, що повний маршрут патрулювання, початок якого надано, є коректним.

Вихідні дані

Виведіть єдине число — тривалість найшвидшого шляху до входу в будинок у секундах. Якщо дістанися входу так, щоби робот-охоронець не помітив Микиту, неможливо, виведіть число -1 .



Приклад

Вхідні дані	Вихідні дані
<pre>8 6 16 ##.##.## 1 1 1 8 5 3 SSRRRSSRLLLLLS</pre>	14
<pre>5 5 4 1 1 5 5 3 3 RDL</pre>	-1
<pre>13 10 64 #####...#######.....##.. .#####.#####. #####.##### ...#.....####. .###.....##.. 9 13 1 7 2 7 RRRRRLLLLLLDDDDDDRRRRRLLLLLLLLLLLLRRRRRUUUUUUUULLLLLLRRRR</pre>	89

Ви можете подивитися анімації, що показують один із можливих шляхів Микити, за наведеними нижче посиланнями. Порада: якщо поставити анімацію на паузу, то за допомогою клавіші «.» (крапка) можна перемикається на наступний кадр.

1. <https://asciinema.org/a/6SdzjgTOMhvGUH0W0bajc3agq>
2. <https://asciinema.org/a/MKVLWWEgGnrVKNIV50jZkdZBN>
3. <https://asciinema.org/a/SeEetraPWnzci52hDz0HaXFRV>



H. Labyrinth

Ограничения: время работы — 1 секунда, использование памяти — 256МБ

Как известно, для соревнования роботов используются различные трассы, которые должен пройти робот за некоторое время. Также роботы часто соревнуются на сообразительность путем прохождения лабиринта. Организаторы соревнований для упрощения подготовки полигона решили изготовить его на основе кубических модулей, выстраивая с их помощью стены и проходы лабиринта. Однако, перед тем как построить лабиринт было решено рассмотреть различные его проекты и выбрать из них наиболее интересные.

Рассмотрим лабиринт в нескольких уровнях в виде «горы». Всё поле в таком лабиринте разделяется на квадраты по количеству $M \times N$, которые по размерам соответствуют кубам, из которых изготавливается конструкция. На плане лабиринта каждый квадрат обозначается при помощи количества уровней, занятых кубами (кубы ставятся один на другой), если в соответствующем квадрате они присутствуют, или 0 — в противном случае. Робот в таком лабиринте может перемещаться в четырех направлениях в горизонтальной плоскости — влево, вправо, вперед и назад (по диагонали робот не перемещается). Также он может перемещаться в вертикальной плоскости вверх или вниз с перемещением на соседнюю клетку, если соседние клетки поля различаются между собой по высоте на один уровень. Вход в лабиринт — слева сверху (по координатам $X=0, Y=0$), выход — справа внизу (по координатам $X=M, Y=N$).

Пожалуйста, помогите организаторам и составьте программу проверки возможности прохождения лабиринта роботом (наличия пути из точки с координатами $(0,0)$ в (M,N)) и вычисления минимального необходимого на это времени. Считая при этом скорость перемещения в горизонтальной плоскости на соседнюю клетку за постоянную L , а в вертикальной с перемещением на соседнюю клетку — в K раз медленнее.

Входные данные

В первой строке находятся четыре целых числа: M ($2 \leq M \leq 100$), N ($2 \leq N \leq 100$), L ($1 \leq L \leq 1000$) и K ($1 \leq K \leq 100$), разделенные пробелами. Затем следует N строк по M целых чисел H_{ij} ($0 \leq H_{ij} \leq 100$), также разделенных пробелами. Они описывают матрицу расположения препятствий — количество кубиков в соответствующей клетке.

Выходные данные

Выведите минимальное время, необходимое для прохождения роботом лабиринта, или -1 , если прохождение невозможно.

Пример

Входные данные	Выходные данные
4 5 1 5 0 2 1 1 1 2 1 0 3 3 4 2 2 4 1 3 1 1 0 0	33
4 5 2 7 0 1 2 3 7 6 5 4 7 7 7 6 2 3 4 5 1 1 4 0	-1