

А. Міжпланетний інтернет 2

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

Надворі 2225 рік. Рома Лонгінт навчається на Землі в X-OSPU (eXtreme Open Source Programming University). На жаль, за двісті років курсові роботи так ніхто й не скасував, тому зараз Рома працює над курсовою з предмету «Мережеві технології». Завданням курсової є побудова мережі міжпланетного зв'язку в одній із планетних систем у власній тренувальній галактиці X-OSPU.

Як відомо, будь-яку планетну систему можна уявити у вигляді ланцюжка планет (починаючи з найближчої до зірки), вважаючи відстань між ними однаковою. Для побудови мережі міжпланетного зв'язку на деяких планетах встановлюють квантові антени великого радіусу дії, які здатні забезпечити мережеве покриття одразу на всій своїй та на кількох сусідніх планетах. Згідно зі стандартами міжпланетного зв'язку кожна квантова антена випромінює квантові хвилі одного з двох кольорів (довжин хвилі): зачарованого (charmed) або дивного (strange). Для забезпечення покриття на планеті, до неї мають дійти хвилі одного, визначеного, із цих двох кольорів, залежно від особливостей її атмосфери.

Квантові хвилі розповсюджуються від планети з антеною в обидва боки вздовж ланцюжка планет на необмежену відстань, але, коли хвилі різних кольорів зустрічаються, вони деструктивно інтерферують і далі не розповсюджуються. Така зустріч завжди стається рівно посередині між двома антенами, що випромінюють ці хвилі. Якщо хвилі різних кольорів зустрічаються не між планетами, а на одній із планет, то через деструктивну інтерференцію на цій планеті не буде мережевого покриття. Коли зустрічаються квантові хвилі одного кольору, жодної інтерференції не відбувається.

Рома вже проаналізував властивості атмосфери всіх планет зі своєї планетної системи та визначив, квантові хвилі якого кольору повинні дійти до кожної з них. Допоможіть йому вирішити, на яких планетах слід встановити квантові антени. Знайдіть мінімальну кількість квантових антен, необхідну для того, щоб забезпечити мережеве покриття на всіх планетах у заданій планетній системі.



Вхідні дані

Дано рядок довжиною від 1 до 10^5 символів, що складається з літер «C» і «S». Кожен символ в цьому рядку описує одну планету і вказує колір квантових хвиль (charmed або strange), які повинні дійти до цієї планети, щоб на ній було забезпечено мережеве покриття.

Вихідні дані

Виведіть єдине число — мінімальну необхідну кількість квантових антен.

Приклади

Вхідні дані	Вихідні дані
CCCCC	1
CSCSC	5
CCSSSCCCCCS	5

У третьому прикладі квантові антени слід встановити на виділених планетах: **CCSSSCCCCCS**.

B. Christmas lights

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

Перед Різдвом сім'я вирішила прикрасити стіну новорічною гірляндою. Вони хочуть закріпити її у двох крайніх точках вздовж верхнього краю стіни (на межі стіни та стелі). Якщо потрібно, вони можуть додавати додаткові кріплення тільки вздовж цього самого краю, тобто всі кріплення мають бути на одній висоті. Через особливості гірлянди, вона провисає між сусідніми кріпленнями у вигляді рівнобедрених трикутників.

Ваша задача — визначити мінімальну кількість додаткових кріплень (окрім двох кутових), щоб гірлянда не торкалася підлоги.



Вхідні дані

Дано три цілих числа:

- W ($1 \leq W \leq 100$) — ширина стіни (в метрах)
- H ($1 \leq H \leq 100$) — висота стіни (в метрах)
- L ($1 \leq L \leq 10^8$) — загальна довжина гірлянди (в метрах)

Також гарантується, що $H \leq W \leq L$.

Вихідні дані

Одне ціле число — мінімальну кількість додаткових кріплень, необхідних для того, щоб гірлянда не торкалася підлоги.

Приклади

Вхідні дані	Вихідні дані
4 3 10	1

У цьому прикладі, якщо не додати жодного кріплення, то гірлянда провисне на 4.58 метра. Це більше, ніж висота стіни (3 метри), тому вона торкнеться підлоги.

C. ElectroTransport

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

Є схема міського електротранспорту (трамваїв та тролейбусів), що зв'язує між собою деяку кількість станцій (N). Вартість проїзду між станціями різна. Рух між станціями можливий в обох напрямках. Поїздка на одному виді транспорту можлива лише між двома станціями (на одну зупинку). Необхідно побудувати маршрут переміщення між початковим пунктом і кінцевим таким чином, щоб види транспорту чергувалися між собою (трамвай-тролейбус-трамвай-тролейбус-... або тролейбус-трамвай-тролейбус-...).



Вхідні дані

У першому рядку задається кількість станцій N ($2 \leq N \leq 100$) і маршрутів M ($1 \leq M \leq 1000$), номер початкової станції A ($0 \leq A < N$) та кінцевої станції B ($0 \leq B < N$). Далі в кожному з наступних M рядків — по чотири числа: номери двох зв'язаних маршрутами станцій C_i та D_i ($0 \leq C_i, D_i < N$), вид транспорту (0 — трамвай або 1 — тролейбус) та вартість проїзду S_i (ціле, $0 \leq S_i \leq 10000$).

Вихідні дані

Одне ціле число — мінімальна вартість проїзду між двома станціями A та B або -1 у разі неможливості проїзду.

Приклади

Вхідні дані	Вихідні дані
<pre> 5 7 0 4 0 1 0 10 0 2 1 5 1 2 0 2 1 3 1 1 2 3 0 15 2 4 1 10 3 4 0 5 </pre>	13

Д. Послідовні числа

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

Індіана Джонс на руїнах календаря майя знаходить рядок із розбірливих і нерозбірливих цифр та здогадується, що це записані поспіль без пробілів послідовні десяткові додатні числа. Яке мінімальне стоїть на початку?

Вхідні дані

В єдиному рядку входу записано рядок довжиною до 1000 символів: цифр та знаків питання. Гарантується, що відповідь існує.

Вихідні дані

Виведіть мінімальне ціле додатне число, з якого може починатися знайдений запис.



Приклади

Вхідні дані	Вихідні дані
10111213	10
1011?214	10110214
?????	4
?1	11
???1??0	999

Примітка

Індіану Джонса безперечно чекає вбивчий скепсис на кафедрі: звідки б майя дізналися арабські цифри?

Е. AI замінить математиків

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

Надворі 2225 рік. Рома Лонгінт навчається на Землі в X-OSPU (eXtreme Open Source Programming University). Прилетівши до університету з Марса на захист курсової роботи, Рома перш за все вирушив до центру квантового копіювання, щоб роздрукувати голограми. Прийшовши до копії-центру, він побачив у черзі робота нової моделі (як ви пам'ятаєте, минулого року в X-OSPU дозволили вступати роботам із просунутим штучним інтелектом). Оскільки чекати потрібно було довго, Рома вирішив познайомитися з ним.

— Привіт! Ти — робот-програміст нової моделі? — запитав Рома.

— Ні, — відповів робот, — я робот-математик моделі M47h-bot.

— Дуже приємно познайомитися. Скажи, а ти зможеш обчислити, скільки буде 4613809953 помножити на 5477491903?

— Легко, — миттєво відповів робот, — результат буде 91332183079596273935.

— Ого, так швидко! — вразився Рома.

Тут їхньою розмовою зацікавився один зі студентів-людей, який також стояв у черзі.

— А яке число більше, 9.9 або 9.11? — запитав студент.

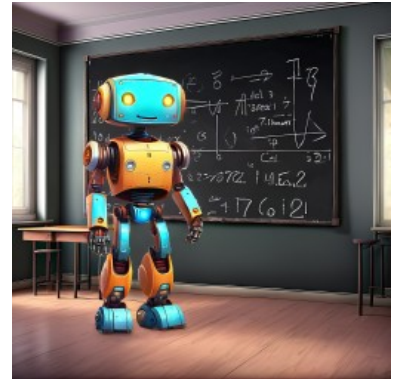
— Хм, — задумався M47h-bot. — Мій пра-пра-пра-пра-дідусь, якому зараз майже двісті років, розповідав мені, що це — дуже складна задача. Давайте міркувати крок за кроком. У нас є два дробових числа, 9.9 та 9.11. Спочатку давайте порівнюємо їх цілі частини. Якщо ціла частина одного числа більша, то це число є більшим. У числі 9.9 ціла частина — 9. У числі 9.11 ціла частина — 9. Очевидно, що 9 дорівнює 9. Отже, цілі частини рівні. Тоді потрібно порівняти дробові частини. Якщо дробова частина одного числа більша, то це число є більшим, інакше — числа є рівними. У числі 9.9 дробова частина — 9. У числі 9.11 дробова частина — 11. Як ми знаємо, число 11 більше за число 9. Отже, число 9.11 більше за число 9.9.

Тут усі студенти-люди, що стояли в черзі, почали сміятися.

— Використай Python, — шепнув Рома роботу.

— Хм, гарна ідея! — сказав M47h-bot.

Допоможіть M47h-bot якнайшвидше реалізувати **його алгоритм** порівняння дробових чисел (будь-якою мовою програмування).



Вхідні дані

Дано два дробових числа, розділених пробілом. Числа містять від одного до п'яти розрядів перед і від одного до п'яти розрядів після десяткової крапки. Числа можуть бути від'ємними.

Вихідні дані

Якщо, згідно з алгоритмом M47h-bot, одне із заданих чисел більше, ніж інше, виведіть відповідь у форматі «Number <число_1> is greater than <число_2>». Інакше, виведіть відповідь у форматі «Numbers <число_1> and <число_2> are equal». Числа слід виводити точно в тому вигляді, в якому вони подані на вході (з усіма незначущими нулями, якщо вони є).

Приклади

Вхідні дані	Вихідні дані
9.9 9.11	Number 9.11 is greater than 9.9
9.9 9.09	Numbers 9.9 and 9.09 are equal
9.9 9.90	Number 9.90 is greater than 9.9
-9.9 -9.11	Number -9.11 is greater than -9.9

Г. Усихання грибів (завдання від хіміків)

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

У грибах вагою W кг вологість становить H_1 процентів. Якою буде вага грибів при усиханні, тобто при зменшенні вологості до H_2 процентів?

Вхідні дані

У єдиному рядку дано три цілих числа, розділених пробілами: W ($1 \leq W \leq 2000$) — початкова вага грибів до усушки, H_1 ($2 \leq H_1 \leq 99$) — початковий відсоток вологості (в процентах), H_2 ($1 \leq H_2 < H_1$) — результуючий відсоток вологості (в процентах).

Вихідні дані

Виведіть єдине число — результуючу вагу грибів після усихання з точністю 4 знаки після коми.

Приклади

Вхідні дані	Вихідні дані
100 99 97	33.3333
100 99 96	25.0000

G. Driving exam

Обмеження на час роботи: для C, C++ і Python — 1 секунда, для Java і Kotlin — 2 секунди
Обмеження на використання пам'яті: 256 МБ

Учень автошколи намагається записатися на складання практичного іспиту. Для цього йому потрібен талон — перепустка на екзамен, що доступна лише онлайн. Однак талони швидко зникають, і отримати бажану дату вкрай складно. Щоб не пропустити можливість, учень вирішує періодично перевіряти сайт щодо наявності талонів. Протягом дня на сайті з'являються та зникають талони. Кожен талон доступний у певному часовому інтервалі $[x_i, y_i]$, де:

- x_i — момент часу (у секундах від початку дня), коли талон з'являється
- y_i — остання секунда, коли талон ще доступний (у момент y_i+1 він уже зникає)

Загальна кількість талонів N задовольняє $1 \leq N \leq 10^5$. Час вимірюється в секундах від 1 до $24 \times 60 \times 60 = 86400$. Учень перевіряє сайт через рівні проміжки часу, тобто якщо він вибирає періодичність P , то перевірки виконуються в моменти: $P, 2P, 3P, \dots$

Вам потрібно знайти найменше можливе значення P , при якому учень гарантовано не потрапить на жоден доступний талон.

Вхідні дані

У першому рядку дано ціле число N ($1 \leq N \leq 10^5$) — кількість талонів. Далі дано N рядків, кожен з яких містить по два числа x_i та y_i ($1 \leq x_i \leq y_i \leq 86400$) — проміжок часу, коли доступний i -й талон.

Вихідні дані

Виведіть число P — період, при якому учень не знайде жодного талону, або -1 , якщо такої відповіді не існує на проміжку $[1; 86400]$.

Приклади

Вхідні дані	Вихідні дані
4 4 6 8 11 15 19 25 27	7

Якщо учень перевіряє сайт з періодичністю $P=7$, він робить перевірки в моменти: 7, 14, 21, 28, Жодна з цих секунд не потрапляє в жоден інтервал талонів, тому 7 є правильною відповіддю.

Н. Турова в'язниця

Обмеження: час роботи — 2 секунди, використання пам'яті — 256 МБ

У Каїссляндії проголосили республіку і всіх королів відправили на заслання. Королям видали окрему нескінченну шахову дошку, а щоб вони між собою не перетиналися, її стережуть N тур із координатами (X_i, Y_i) . Король не може ні пройти через бите турою поле, ні взяти саму туру — у них той самий колір. Скільки королів можна помістити на таку дошку, щоб вони не могли стати на сусідні клітини один з одним?

Король ходять у одну з 8 сусідніх клітин, не може ставати на поле з турою, або на поле, яке бить турою. Тура б'є по горизонталі та вертикалі на будь-яку кількість клітин.

Вхідні дані

У першому рядку записано число N (ціле, $0 \leq N \leq 2 \cdot 10^5$). У наступних N рядках записані через пробіл пари координат X_i та Y_i (цілі, $-10^6 \leq X_i, Y_i \leq 10^6$). Усі пари різні.

Вихідні дані

Виведіть єдине ціле невід'ємне число — максимальну кількість королів, яку можна поставити на нескінченну дошку так, щоб жодні їхні допустимі ходи не привели жодного з них на сусідню клітку з іншим королем.

Приклади

Вхідні дані	Вихідні дані
0	1
1 0 0	4
4 1 1 2 1 4 1 4 3	9

Пояснення до третього прикладу:

