

А. Повний чіназес

Обмеження: 2 сек., 512 МіБ

Зеник закінчив змагатись в ІСРС, випустився з університету, і влаштувався на роботу. Інколи він заробляє гроші, а інколи витрачає їх. Наразі на рахунку Зеника 0 гривень.

Всього Зеник здійснив n дій, в i -ту дію він або заробив a_i гривень, або витратив a_i гривень.

Якщо в якийсь момент часу баланс Зеника був від'ємним, то він був у боргу. Інакше, на думку Зеника, відбувся повний чіназес. Допоможіть Зенику визначити, чи був його баланс від'ємним!

Вхідні дані

Перший рядок містить одне ціле число n . i -ий з наступних n рядків може мати один з наступних двох форматів:

- `earn a_i` , що означає, що Зеник заробив a_i гривень;
- `spend a_i` , що означає, що Зеник витратив a_i гривень.

Вихідні дані

Якщо Зеник не був у боргу протягом даних n дій, в єдиному рядку виведіть `chinazes`. Інакше, виведіть `debt`.

Обмеження

- $1 \leq n \leq 100$,
- $1 \leq a_i \leq 100$,
- a_i — цілі числа.

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
3 earn 5 spend 6 earn 5	debt
4 earn 69 spend 42 earn 42 spend 69	chinazes

Примітки

У першому прикладі, баланс Зеника після першої дії: 5, після другої: -1 , після третьої: 4. Незважаючи на те, що його баланс в кінці невід'ємний, після другої дії Зеник був у боргу, тому ми маємо вивести `debt`.

У другому прикладі, баланси Зеника після даних 4 дій: $[69, 27, 69, 0]$; вони ніколи не були від'ємними.

В. Менші зліва, менші справа

Обмеження: 2 сек., 512 МіБ

Організатори змагань задали масив a_i з n елементів.

Для кожного елемента відомо значення l_i — кількість менших елементів зліва від нього та r_i — кількість менших елементів справа. Знайдіть будь-який масив, який підходить під дані обмеження.

Для заданих вхідних даних гарантовано існує хоча б один масив. Якщо існує декілька масивів, які задовільняють умови, виведіть будь-який з них.

Вхідні дані

У першому рядку задано єдине ціле число n — довжину масиву.

У наступних n рядках задано по 2 цілих числа — l_i та r_i .

Вихідні дані

У єдиному рядку виведіть n чисел — значення елементів масиву. Усі значення повинні бути в межах від 0 до 10^9 .

Якщо існує декілька масивів, які задовільняють умови, виведіть будь-який з них.

Обмеження

$$1 \leq n \leq 10^5,$$

$$0 \leq l_i, r_i \leq n - 1.$$

Гарантується, що для заданих вхідних даних існує хоча б один масив.

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
4 0 0 1 2 1 1 0 0	4 47 7 4
4 0 0 0 0 0 0 0 0	4 4 4 4

С. Потрійний XOR

Обмеження: 2 сек., 512 МiБ

Вам дані два натуральні числа $m < n$. Яку найбільшу кількість попарно різних цілих чисел можна обрати в інтервалі $[m, n]$ так, щоб виконувалась наступна умова:

- Серед обраних чисел немає таких трьох різних чисел x, y, z , що $x \oplus y \oplus z = 0$.

Нагадаємо, що \oplus позначає операцію побітового виключного АБО. Наприклад, $13 \oplus 6 = 11$, адже в двійковому записі $13 = 1101$, а $6 = 0110$, тому їх \oplus має бути рівним $1011 = 11$.

Вхідні дані

Єдиний рядок вхідних даних містить два натуральні числа m, n .

Вихідні дані

У першому рядку виведіть число k — максимальну кількість чисел, яку можна обрати, задовольняючи умову задачі.

У другому рядку виведіть k попарно різних цілих чисел a_1, a_2, \dots, a_k , $m \leq a_i \leq n$. Вони мають задовольняти умові задачі.

Якщо існує кілька різних способів обрати максимальну можливу кількість чисел, знайдіть будь-який.

Обмеження

$$1 \leq m < n \leq 2 \cdot 10^5.$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
1 7	4 1 3 5 7
2024 2025	2 2025 2024

Примітки

У першому прикладі можна показати, що вибрати 5 чисел з інтервалу $[1, 7]$ таким чином не можна.

У другому прикладі серед $[2024, 2025]$ нема трьох різних чисел, тому умова виконується автоматично.

Д. Лабіринт

Обмеження: 4 сек., 1024 МiБ

Зеник знаходиться в лабіринті, який можна представити як дерево з n вершинами та $n - 1$ ребрами, де вершини — це кімнати, а ребра — проходи між ними. З кожної кімнати можливо добратися в кожен іншу використовуючи переходи між кімнатами. Вершина 1 є виходом з лабіринту, до якого Зеник прагне потрапити якомога швидше.

Кожну секунду всі ребра незалежно та рівномірно отримують нову орієнтацію. Для кожного ребра між двома суміжними вершинами u та v ребро стає орієнтованим або з u в v , або з v в u , з ймовірністю $\frac{1}{2}$ кожне. Ребра ніколи не бувають двонаправленими; вони завжди орієнтовані лише в один бік.

Після зміни орієнтації ребер Зеник знає поточну орієнтацію всіх ребер і приймає рішення про свій наступний крок з метою мінімізувати очікуваний час досягнення вершини 1.

Зеник діє наступним чином:

- Якщо з його поточної вершини v існує принаймні одне ребро, орієнтоване з v в сусідню вершину u , він повинен обрати одну з таких вершин та переміститися до неї. Переміщення займає 1 секунду.
- Якщо з вершини v немає жодних вихідних ребер, Зеник змушений чекати 1 секунду до наступної переорієнтації ребер.

Орієнтація ребер змінюється щосекунди, одразу перед тим, як Зеник приймає рішення про свій наступний крок.

Ваша задача — допомогти Зенику визначити для кожної вершини v ($2 \leq v \leq n$) математичне очікування часу (у секундах), необхідного для досягнення вершини 1, починаючи з вершини v , за умови, що він діє оптимально, щоб мінімізувати цей час.

Вхідні дані

Перший рядок містить одне ціле число n — кількість вершин у дереві.

i -ий з наступних $n - 1$ рядків містить два цілих числа u_i та v_i , що означає, що між вершинами з цими індексами є ребро.

Вихідні дані

Виведіть $n - 1$ чисел. k -те число у рядку має означати математичне очікування для вершини $k + 1$.

Відповідь буде зарахована, якщо її абсолютна чи відносна похибка не буде більшою ніж 10^{-7} .

Обмеження

$$2 \leq n \leq 2 \cdot 10^5,$$
$$1 \leq u_i, v_i \leq n.$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
4 1 2 2 3 2 4	3.5000000000 5.5000000000 5.5000000000
2 1 2	2.00000000

Е. Трикутники на площині

Обмеження: 2 сек., 512 МіБ

На площині задано $3 \cdot n$ точок так, що жодні три точки не лежать на одній прямій. Потрібно побудувати n трикутників таким чином, щоб виконувались такі умови:

1. Кожна точка є вершиною лише одного трикутника.
2. Для будь-яких двох трикутників площа їхнього перетину дорівнює 0.

Можна показати, що за обмежень задачі це завжди можливо.

Вхідні дані

У першому рядку задано одне ціле число n — кількість трикутників, які необхідно побудувати.

У наступних $3 \cdot n$ рядках задано по два цілих числа x_i, y_i — координати точок.

Вихідні дані

Виведіть n рядків, кожен з яких містить три числа — індекси точок, які утворюють відповідний трикутник.

Якщо існує декілька можливих способів утворити n трикутників, виведіть будь-який.

Обмеження

$$1 \leq n \leq 500,$$

$$1 \leq x_i, y_i \leq 10^9.$$

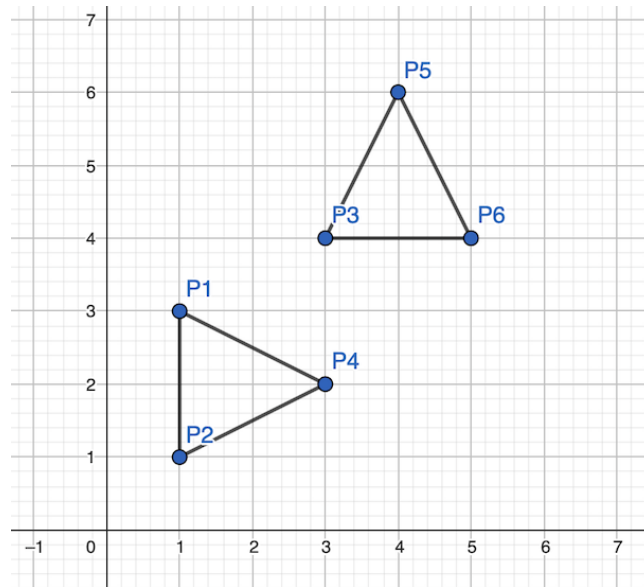
Всі точки обов'язково попарно різні.

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
2	1 2 4
1 3	3 5 6
1 1	
3 4	
3 2	
4 6	
5 4	

Примітки

Рис. 1: Перший приклад.



Г. Обміни з максимальною сумою

Обмеження: 2 сек., 512 МіБ

Вам задано масив a_1, a_2, \dots, a_{2n} довжиною $2n$. За одну операцію можна вибрати довільний індекс i з $1 \leq i \leq 2n - 1$ і поміняти місцями a_i, a_{i+1} .

Для кожного k від 0 до n^2 знайдіть найбільшу можливу суму $a_1 + a_2 + \dots + a_n$, яку можна отримати після виконання не більше k операцій.

Вхідні дані

Перший рядок містить єдине ціле число n .

Другий рядок містить $2n$ цілих чисел a_1, a_2, \dots, a_{2n} — елементи масиву.

Вихідні дані

Виведіть $n^2 + 1$ ціле число: найбільшу можливу суму $a_1 + a_2 + \dots + a_n$, яку можна отримати, виконавши не більше $0, 1, 2, \dots, n^2$ операцій.

Обмеження

$$1 \leq n \leq 100,$$

$$1 \leq a_i \leq 10^9.$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
3 1 2 3 4 5 6	6 7 8 9 10 11 12 13 14 15

Примітки

Якщо не виконати жодної операції, сума $a_1 + a_2 + a_3 = 1 + 2 + 3 = 6$.

За одну операцію можна поміняти місцями a_3, a_4 , отримавши $a_1 + a_2 + a_3 = 1 + 2 + 4 = 7$.

G. Прямокутник і точки

Обмеження: 2 сек., 512 MiB

Дано прямокутник розміром $n \times m$ на координатній площині. Лівий нижній кут прямокутника знаходиться в точці $(0, 0)$, а правий верхній — в точці (n, m) . Всередині цього прямокутника розташовано k точок.

Потрібно перемістити прямокутник (не обертаючи і не змінюючи його розмірів) так, щоб всі k точок опинилися на межах або поза межами прямокутника. Переміщення може здійснюватися у будь-якому напрямку на площині. Відстань, на яку перемістили прямокутник, визначається як евклідова відстань між початковим положенням лівого нижнього кута $(0, 0)$ та новим положенням цього кута (dx, dy) .

Ваше завдання — знайти мінімальну відстань, на яку потрібно перемістити прямокутник, щоб кожна з k точок опинилася на межах або поза межами прямокутника.

Вхідні дані

Перший рядок містить три цілі числа n , m , k — розміри прямокутника та кількість точок всередині нього.

Наступні k рядків містять по два цілі числа x_i , y_i — координати точок, розташованих всередині прямокутника.

Вихідні дані

Виведіть одне число — мінімальну евклідову відстань, на яку потрібно перемістити прямокутник, щоб усі k точок опинилися поза його межами.

Обмеження

$$\begin{aligned} 2 &\leq n, m \leq 10^9, \\ 1 &\leq k \leq 2 \cdot 10^5, \\ 1 &\leq x_i < n, 1 \leq y_i < m. \end{aligned}$$

Приклади

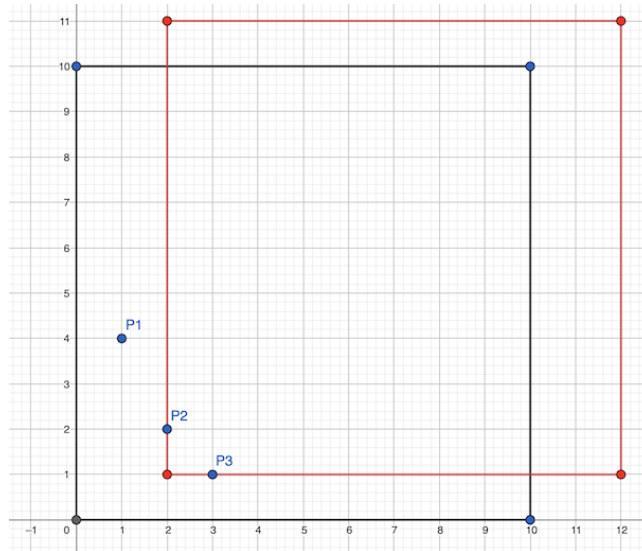
Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
10 10 3 1 4 2 2 3 1	2.2360679775

Примітки

Відстань між двома точками (x_1, y_1) та (x_2, y_2) обчислюється за формулою $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

Червоним кольором показано прямокутник після переміщення, його лівий нижній кут знаходиться в точці $(2, 1)$. Тому відстань на яку було пересунуто рахуємо як $\text{dist}((0, 0), (2, 1)) = \sqrt{(2 - 0)^2 + (1 - 0)^2} = 2.2360679775$.

Рис. 2: Перший приклад.



Н. Доставка їжі

Обмеження: 2 сек., 512 MiB

Зеник скористався своїм улюбленим сервісом доставки їжі Slovo. Всього хлопець замовив на a гривень. Згідно з правилами компанії Slovo, доставка безплатна, якщо замовлення на не менше ніж b гривень, інакше вартість доставки – c гривень.

Допоможіть Зенику порахувати, скільки ж він має заплатити всього.

Вхідні дані

У єдиному рядку задано 3 цілих числа a , b , c – сума замовлення, мінімальна сума для безкоштовної доставки та вартість доставки відповідно.

Вихідні дані

Виведіть єдине число – скільки ж повинен заплатити Зеник.

Обмеження

$$1 \leq a, b \leq 5000,$$

$$1 \leq c \leq 100.$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
744 1000 74	818
1477 1000 74	1477

Примітки

У першому прикладі Зеник замовив менше ніж на 1000 гривень, тож йому потрібно оплатити також доставку.

У другому прикладі доставка безплатна.

I. Перший етап

Обмеження: 2 сек., 512 МіБ

У першому етапі одного цілком вигаданого змагання беруть участь n команд. Та, на жаль, правила проходження у другий етап занадто заплутані, тож допоможіть організаторам визначити, скільки ж команд потрапить у другий етап.

У другий етап проходять:

- Команди, які потрапили у топ m команд загальної таблиці, проте **не більше ніж** k команд з одного університету;
- Хоча б одна команда з кожного регіону, навіть якщо вона не потрапила у топ m .

Дано результати першого етапу у порядку від найкращої до найгіршої команди. Визначте, скільки команд потрапить у другий етап. Жодні 2 чи більше команди не можуть ділити те ж місце.

Вхідні дані

В першому рядку задано 3 числа n , m та k – кількість команд, топ команд, які проходять у другий етап, та максимальна кількість команд з одного університету.

В наступних n рядках задано по 3 рядки розділених пробілами – назва команди, назва регіону та назва університету. Команди задані в порядку від першої до останньої.

Вихідні дані

Виведіть єдине число – кількість команд, які потраплять у другий етап.

Обмеження

$$1 \leq n \leq 1000,$$

$$1 \leq m \leq 200,$$

$$1 \leq k \leq 5.$$

Усі назви складаються з великих чи маленьких латинських букв, цифр чи символу `_` довжиною до 20 символів. Назви є різними, якщо відрізняються хоча б одним символом, маленькі та великі букви вважаємо різними.

Усі назви команд різні, команди з одного університету також представляють один регіон.

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
<pre>7 4 1 KNU_0_GB_RAM Kyiv KNU LNU_Stallions Lviv LNU KhNURE_KIVI Kharkiv KhNURE KNU_Banderolki Kyiv KNU UzhNU_Machata Zakarpattia UzhNU LNU_Zenyk47_Beer Lviv LNU UzhNU_TroubleMakers Zakarpattia UzhNU</pre>	4
<pre>7 4 2 team1 region university team2 region university team3 region university team4 region university team5 region university team6 region university team7 region university</pre>	2
<pre>7 4 2 team1 region university1 team2 region university2 team3 region university3 team4 region university4 team5 region university5 team6 region university6 team7 region university7</pre>	4
<pre>7 4 2 team1 region1 university1 team2 region2 university2 team3 region3 university3 team4 region4 university4 team5 region5 university5 team6 region6 university6 team7 region7 university7</pre>	7
<pre>2 7 4 team1 region1 university1 team2 region2 university2</pre>	2

Примітки

У першому прикладі у другий етап проходять команди `KNU_0_GB_RAM`, `LNU_Stallions`, `KhNURE_KIVI` та `UzhNU_Machata`. Команда `KNU_Banderolki` не пройшла, адже вона зайняла 2 місце серед команд університету `KNU`. Команда `UzhNU_Machata` пройшла, адже вона зайняла перше місце серед команд регіону `Zakarpattia`.

У другому прикладі всі команди з одного університету, отже у другий етап проходять $k = 2$ команди.

У третьому прикладі усі команди з одного регіону, але з різних університетів, отже у другий етап проходять $m = 4$ команд.

У четвертому всі команди з різних регіонів, отже вони усі проходять у другий етап.

Ж. Гільдія злодіїв

Обмеження: 3 сек., 512 МіБ

В одному королівстві є n міст, з'єднаних $n - 1$ дорогою так, що між кожною парою міст можливо добратись дорогами. Іншими словами, міста та дороги утворюють дерево.

У кожному місті є банк, в сховищі банку в місті v знаходиться a_v монет. В одному з міст зародилась гільдія злодіїв, яка планує пограбувати банки усього королівства. На жаль, король не знає в якому саме місті це сталося, тож для протидії великого пограбування він вирішив зробити усі дороги королівства односторонніми, попри те, що після цього уже неможливо дістатись з будь-якого міста в будь-яке інше.

Та все ж, пограбування відбудеться! Гільдія пограбує усі банки в усіх містах, в які можливо добратись від міста, де гільдія зародилась.

Король хоче обрати такі напрямки доріг, щоб мінімізувати максимальний можливий збиток. Більш формально, нехай c_v – сума кількості монет у всіх містах, які досяжні з міста v після того, як король обере напрямок кожної дороги (включно з містом v). Тоді вам потрібно обрати напрямки так, щоб мінімізувати $\max c_v$.

Вхідні дані

В першому рядку задано єдине ціле число n – кількість міст королівства.

У наступному рядку задано n цілих чисел a_i – кількість монет у банках.

i -ий з наступних $n - 1$ рядків містить два цілі числа u_i та v_i , що означає, що між містами з цими номерами є дорога.

Вихідні дані

Виведіть єдине число – мінімальне можливе значення $\max c_v$.

Обмеження

$$2 \leq n \leq 10^5,$$

$$1 \leq a_i \leq 10^9,$$

$$1 \leq u_i, v_i \leq n.$$

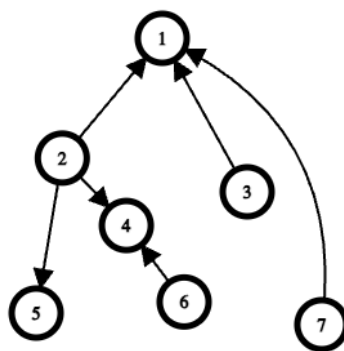
Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
<pre>7 1 2 3 4 5 6 7 1 2 1 3 2 4 2 5 4 6 1 7</pre>	12
<pre>7 1 1 1 1 1 1 1 1 2 1 3 2 4 2 5 4 6 1 7</pre>	3
<pre>2 4 7 1 2</pre>	11

Примітки

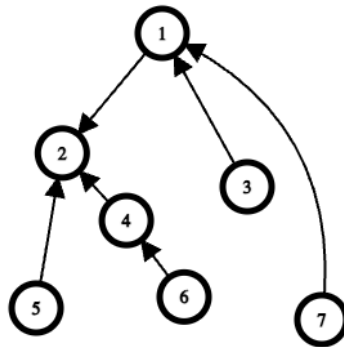
У першому прикладі у вершині v знаходиться v монет. Можлива орієнтація ребер зображена на рисунку. Тоді якщо гільдія розташована у вершині 2, а отже вони можуть пограбувати банк у вершинах 1, 2, 4 та 5. $c_2 = a_1 + a_2 + a_4 + a_5 = 12$. Для всіх інших вершин це значення є меншим.

Рис. 3: Перший приклад.



У другому прикладі у кожній вершині по одній монеті. Один з варіантів, як орієнтувати ребра:

Рис. 4: Другий приклад.



К. Зростаюча таблиця

Обмеження: 2 сек., 512 МіБ

Вам дані натуральні числа n, m , і $n \cdot m$ цілих чисел a_1, a_2, \dots, a_{nm} . Чи можна їх розставити в таблиці $n \times m$ таким чином, щоб:

- Числа в кожному рядку йшли в порядку зростання, зліва направо;
- Числа в кожному стовпчику йшли в порядку зростання, зверху вниз?

Якщо так, виведіть приклад відповідно заповненої таблиці.

Вхідні дані

Перший рядок вхідних даних містить одне натуральне число t — кількість тестових наборів.

Перший рядок кожного тестового набору містить два натуральні числа n, m — розміри таблиці.

Другий рядок кожного тестового набору містить $n \cdot m$ натуральних чисел a_1, a_2, \dots, a_{nm} .

Вихідні дані

Для кожного тестового набору, якщо відповідним чином заповнити таблицю неможливо, виведіть NO.

Інакше, виведіть YES. В i -му з наступним n рядків виведіть m чисел $b_{i,1}, b_{i,2}, \dots, b_{i,m}$ — елементи i -го рядка.

Якщо існує кілька способів заповнити таблицю таким чином, виведіть будь-який.

Обмеження

$$1 \leq t \leq 10000,$$

$$1 \leq n \cdot m \leq 2 \cdot 10^5,$$

$$1 \leq a_i \leq 10^9,$$

Сума $n \cdot m$ по всіх тестовим наборах не перевищує $2 \cdot 10^5$.

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
3	YES
1 4	1 2 3 4
4 3 2 1	NO
2 3	YES
1 1 2 2 3 3	1 2
3 2	2 3
1 2 2 3 3 4	3 4

Примітки

Приклади для першого та третього тестового набору наведені в вихідних даних. Можна показати, що для другого тестового набору розташувати числа задовольняючи умові задачі неможливо.

Л. Лови

Обмеження: 2 сек., 512 МiБ

Двоє гравців грають гру на площині. Другий гравець хоче зловити першого гравця, в той же час перший гравець хоче досягти одного з безпечних місць. Перший гравець спочатку знаходиться в точці з цілими координатами (x_1, y_1) . Другий гравець у точці з цілими координатами (x_2, y_2) .

За одну секунду гравець переходить в одну з 4 сусідніх точок, тобто з точки (x, y) в одну з точок $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ чи $(x, y - 1)$. Гравці ходять по черзі, першу секунду робить хід перший гравець, наступну секунду другий гравець, потім знову перший і так далі. Перший гравець не може переходити в точку, в якій знаходиться другий гравець.

Безпечними для першого гравця є точки з $x = 0$ або $y = 0$. Перший гравець перемагає, якщо він перейде в одну з таких точок. Другий гравець перемагає, якщо він перейде у точку з другим гравцем, або якщо за 10^{100} ходів гра не закінчиться. В такому випадку гра одразу завершується.

Визначте, чи зможе перший гравець досягти безпечної точки.

Вхідні дані

У першому рядку задано єдине ціле число t – кількість тестових наборів.

У кожному з наступних t рядків задано по 4 цілих числа x_1, y_1, x_2, y_2 – початкові позиції гравців.

Вихідні дані

В кожному з t рядків виведіть YES, якщо перший гравець зможе досягти безпечної точки у відповідному тесті та NO в протилежному випадку.

Обмеження

$$1 \leq t \leq 10^5,$$

$$-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9.$$

Приклади

Вхідні дані (<i>stdin</i>)	Вихідні дані (<i>stdout</i>)
2	YES
1 1 2 2	NO
2 2 1 1	

Примітки

У першому тестовому наборі першому гравцю достатньо піти в одну з точок $(1, 0)$ чи $(0, 1)$ своїм першим ходом, щоб перемогти. На рисунку перший гравець зображений точкою A , другий гравець точкою B , усі безпечні точки позначені червоним.

У другому тесті, якщо перший гравець під вліво або вниз, то другий гравець може зловити його наступним ходом. Якщо перший гравець буде ходити тільки вгору або вправо, то другий гравець може повторювати його ходи поки не досягне ліміту ходів.

Рис. 5: Перший приклад.

