

Задача А. Дерево та особлива вершина

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Задано дерево з n вершин, пронумерованих від 0 до $n - 1$.

Відомо, що з плином часу ребра в дереві можуть зруйнуватись.

Вершина дерева з номером r особлива — з кожної вершини завжди має існувати хоча б один шлях у вершину з номером r .

Передбачається, що сьогодні рівно одне з ребер зруйнується, але невідомо яке саме.

Додайте до графу мінімальну кількість ребер так, щоб при руйнуванні будь-якого ребра, з кожної вершини існував шлях до вершини з номером r .

Формат вхідних даних

У першому рядку записано два цілі числа n, r ($2 \leq n \leq 10^5, 0 \leq r < n$).

У наступних $n - 1$ рядках задано опис ребер дерева у форматі $u v$ ($0 \leq u, v < n, u \neq v$) — номери вершин, які сполучає відповідне ребро.

Формат вихідних даних

У першому рядку вихідного файлу виведіть одне ціле число k — мінімальну кількість ребер, яку потрібно додати до графу, щоб виконувалась необхідна умова.

У наступних k рядках виведіть по два цілі числа u, v ($0 \leq u, v < n, u \neq v$) — номери вершин, які сполучатиме відповідне ребро.

Якщо існує кілька правильних відповідей, дозволяється вивести будь-яку з них.

Приклади

standard input	standard output
4 0 0 1 0 2 0 3	2 3 2 3 1
6 0 0 1 0 2 0 3 1 4 1 5	2 3 5 2 4

Задача В. Лотерея

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2 seconds
Ліміт використання пам'яті:	512 megabytes

Ви вирішили взяти участь в лотерії, де в ряд стоять n скринь з золотом. Ви можете забрати собі будь-яку одну скриню, а тому ви хочете обрати скриню з максимальною кількістю золота.

Для того щоб дізнатися скільки золота містить кожна зі скринь, ви можете купляти підказки у продавців. Всього існує m продавців, i -й з яких торгує c_i підказками. Кожна підказка характеризується числами l_j, r_j та коштує w_j монет. Купівля чергової підказки дозволяє вам дізнатися сумарну кількість золота у скринях з номерами від l_j до r_j .

Для того щоб i -й продавець не залишився ображеним, вам необхідно купити у нього **рівно** k_i ($1 \leq k_i \leq c_i$) підказок.

Знайдіть мінімальну сумарну кількість монет, яку необхідно заплатити для того, щоб можна було однозначно зрозуміти кількість золота у кожній зі скринь.

Формат вхідних даних

У першому рядку знаходиться єдине ціле число t ($1 \leq t \leq 10$) — кількість тестів.

Перший рядок кожного тесту містить два цілі числа n, m ($1 \leq n, m \leq 80$) — кількості скринь та продавців відповідно.

Наступні рядки описують продавців. Перший рядок опису продавця містить два цілі числа c_i, k_i ($1 \leq k_i \leq c_i \leq 80$) — кількість підказок, якими торгує продавець, та кількість підказок, яку необхідно у нього купити. Гарантується, що сумарна кількість підказок у кожному з тестів не перевищує 80 ($\sum_{i=1}^m c_i \leq 80$).

Далі слідує c_i рядків, що описують підказки. У j -му з них знаходяться три цілі числа l_j, r_j, w_j ($1 \leq l_j \leq r_j \leq n, 1 \leq w_j \leq 10^6$) — параметри та вартість підказки.

Формат вихідних даних

Для кожного тесту виведіть єдине ціле число — мінімальну сумарну кількість монет, яку необхідно заплатити для того, щоб можна було однозначно зрозуміти кількість золота у кожній зі скринь. Якщо це зробити неможливо, виведіть -1 .

Приклад

standard input	standard output
2	111
2 2	-1
1 1	
1 2 1	
3 2	
1 1 10	
2 2 100	
1 2 1000	
2 2	
1 1	
1 1 1	
1 1	
1 1 2	

Задача С. Множина

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: `2.5 seconds`
Ліміт використання пам'яті: `256 megabytes`

У вас є пуста мультимножина чисел S . Вам необхідно обробити n запитів. Запити бувають двох типів:

1. $+ x w$ — додати в S число x вагою w .
2. $- x w$ — видалити з S число x вагою w . Гарантується, що перед виконанням цього запиту, число x вагою w було присутнє в мультимножині S .

Після виконання кожного запиту необхідно знайти максимальну сумарну вагу такої підмножини чисел $A \subset S$, що xor елементів будь-якої непорожньої підмножини $B \subset A$ не дорівнює нулю.

Формат вхідних даних

У першому рядку записано єдине ціле число n ($1 \leq n \leq 300\,000$) — кількість запитів.

У наступних n рядках спочатку записано тип чергового запиту c ($c \in \{«+», «-»\}$), після якого записано два цілі числа x, w ($1 \leq x, w \leq 10^9$).

Формат вихідних даних

Виведіть n рядків — максимальну сумарну вагу лінійно незалежної підмножини чисел з S після виконання чергового запиту.

Приклад

standard input	standard output
8	4
+ 1 4	7
+ 1 7	4
- 1 7	9
+ 2 5	9
+ 3 2	14
+ 3 9	13
- 2 5	6
- 3 9	

Задача D. Гра на матроїді

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	1 second
Ліміт використання пам'яті:	256 megabytes

Є набір X , що складається з n предметів, i -й з яких має вагу a_i ($0 \leq i < n$). На цьому наборі побудований матроїд¹ $M = \langle X, I \rangle$, а саме: для кожної підмножини предметів $A \subset X$ відомо, чи є вона незалежною (відомо, чи виконується $A \in I$). Гарантується, що **побудований матроїд є кольоровим**².

Два гравці по черзі додають в спочатку пусту множину S по одному предмету так, щоб множина обраних предметів залишалась незалежною. Гра закінчується коли неможливо додати жоден предмет до множини S так, щоб вона залишилася незалежною. Перший гравець хоче максимізувати сумарну вагу обраних предметів, а другий мінімізувати. Знайдіть сумарну вагу обраних предметів у кінці гри, якщо обидва гравці грають оптимально.

Знайдіть результати m ігор, j -та з яких відрізняється від попередньої вагою предмета з номером p_j . Зверніть увагу, що побудований матроїд M залишається незмінним.

Формат вхідних даних

У першому рядку записано два цілі числа n, m ($1 \leq n \leq 20, 1 \leq m \leq 10^5$) — розмір набору предметів та кількість ігор.

У другому рядку знаходиться 2^n символів c_{mask} ($c_{mask} \in \{0, 1\}$), що описують матроїд. А саме: c_{mask} дорівнює 1 тоді і тільки тоді, коли відповідна множина предметів є незалежною ($A_{mask} \in I$). Нумерація предметів у бітовій масці починається з нуля, а біти записані в порядку від наймолодшого до найстаршого. Наприклад, бітова маска $mask = 5 = 101000\dots 0_2$ позначає множину предметів $\{X_0, X_2\}$. Гарантується, що заданий матроїд є **кольоровим**.

У третьому рядку записано n цілих чисел a_0, a_1, \dots, a_{n-1} ($1 \leq a_i \leq 10^8$) — початкова вага предметів.

Наступні m рядків описують ігри. У j -му з них записано два цілі числа p_j, x_j ($0 \leq p_j < n, 1 \leq x_j \leq 10^8$) — номер предмета та його нова вага.

Формат вихідних даних

Виведіть m рядків — сумарну вагу обраних предметів у кінці кожної гри.

Приклад

standard input	standard output
3 2	15
11101110	105
1 1 5	
0 10	
1 100	

Зауваження

¹Нагадаємо, що у будь-якого матроїда множина I незалежних підмножин предметів задовольняє наступним аксіомам:

1. Пуста множина предметів є незалежною ($\emptyset \in I$).
2. Будь-яка підмножина незалежної множини також є незалежною (якщо $B \in I$ і $A \subset B$, то $A \in I$).
3. Якщо розмір незалежної множини A менше розміру незалежної множини B , то існує хоча б один елемент в B , який можна додати до A так, щоб множина A залишилася незалежною (якщо $A, B \in I$ і $|A| < |B|$, то $\exists x \in B \setminus A, A \cup \{x\} \in I$).

²Нагадаємо що кольоровим називається такий матроїд $M = \langle X, I \rangle$, що усі предмети з X можна пофарбувати в певні кольори так, що кожна множина A предметів з X буде незалежною тоді і тільки тоді, коли усі кольори предметів з A є різними.

Задача Е. Складні задачі

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Нещодавно Олег знайшов цікавий збірник задач. У цьому збірнику є n задач пронумерованих від 1 до n . Оскільки Олег дуже любить прості числа, він вирішив розв'язувати лише ті задачі, номери яких є простими числами.

Але Олег помітив, що кожна задача з простим номером p є доволі складною для нього у випадку, коли $(p^2 - 1)$ не ділиться на 24. Допоможіть Олегу зрозуміти, скільки складних задач доведеться йому розв'язати.

Формат вхідних даних

У першому рядку записано єдине ціле число n ($1 \leq n \leq 0.75 \cdot 10^{10}$) — кількість задач у збірнику Олега.

Формат вихідних даних

Виведіть єдине ціле число — кількість простих чисел p від 1 до n таких, що $(p^2 - 1)$ не ділиться на 24.

Приклади

standard input	standard output
2	1
4	2

Задача F. Відгадай матроїд

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: `3 seconds`
Ліміт використання пам'яті: `256 megabytes`

Є набір X , що складається з n предметів, пронумерованих від 0 до $n - 1$. На цьому наборі предметів журі загадало матроїд¹ $M = \langle X, I \rangle$. Вам необхідно його відгадати. Для цього ви можете задавати питання вигляду «чи є певна підмножина предметів з набору X незалежною». Ви можете задати не більш ніж 50000 питань.

Формат вхідних даних

В першому рядку записано єдине ціле число n ($1 \leq n \leq 18$) — розмір набору предметів.

Формат вихідних даних

Коли ваша програма буде готова вивести відповідь, виведіть строку в форматі «! s », де s — рядок довжини 2^n , s_{mask} ($0 \leq mask < 2^n$) дорівнює 1 тоді і тільки тоді, коли відповідна множина предметів є незалежною. Нумерація предметів у бітовій масці починається з нуля, а біти записані в порядку від наймолодшого до найстаршого. Наприклад, бітова маска $mask = 5 = 101000 \dots 0_2$ позначає множину предметів $\{X_0, X_2\}$. Після надання відповіді програма повинна одразу завершити роботу.

Протокол взаємодії

Для того, щоб задати питання, виведіть строку в форматі «? $mask$ », де $mask$ ($0 \leq mask < 2^n$) означає підмножину предметів з набору X . У відповідь ви повинні прочитати єдине ціле число f ($0 \leq f \leq 1$). Відповідна підмножина предметів є незалежною тоді і тільки тоді, коли $f = 1$.

Якщо ви зробите більш ніж 50000 запитів, ви отримаєте вердикт `Wrong answer`.

Приклад

standard input	standard output
2	? 0
1	? 1
0	? 2
1	? 3
0	! 1010

Зауваження

¹Note, that set I of independent subsets of any matroid meets following axioms:

1. Empty set is independent ($\emptyset \in I$).
2. Any subset of independent set is independent. (if $B \in I$ and $A \subset B$, then $A \in I$).
3. If independent set A has smaller size than independent set B , there exist at least one element in B that can be added into A without loss of independency (if $A, B \in I$ and $|A| < |B|$, then $\exists x \in B \setminus A, A \cup \{x\} \in I$).

Задача G. Остовні дерева

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1.5 seconds
Ліміт використання пам'яті: 256 megabytes

Задано неорієнтований граф з n вершин та m ребер. Необхідно розподілити ребра по k неперетинним групам так, щоб ребра у кожній групі формували дерево на n вершинах. Зверніть увагу, що деякі з ребер можуть не належати жодній з груп.

Формат вхідних даних

У першому рядку записано три цілі числа n, m, k ($2 \leq n, m \leq 200, 1 \leq k \leq 10$) — кількість вершин та ребер в графі, та кількість груп, на яку необхідно розбити ребра.

У наступних m рядках записано по два цілі числа u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) — номери вершин, що з'єднує чергове ребро графу. Зверніть увагу, що у графі можуть бути присутні кратні ребра.

Формат вихідних даних

Якщо розбити ребра графу на k груп неможливо, виведіть єдиний рядок «No».

Інакше у першому рядку виведіть «Yes». Далі виведіть k рядків, що описують групи ребер. Кожен рядок повинен містити $n - 1$ ціле число $num_1, num_2, \dots, num_{n-1}$ ($1 \leq num_i \leq m$) — номери ребер в черговій групі. Нумерація ребер починається з одиниці.

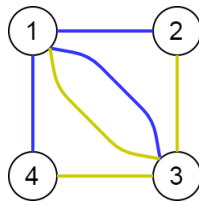
Зверніть увагу, що кожен номер num_i ($1 \leq num_i \leq m$) може бути присутнім не більш ніж у одній групі, та у кожній групі описані ребра повинні формувати дерево на n вершинах.

Приклади

standard input	standard output
4 5 1 1 2 2 3 1 3 2 4 3 4	Yes 1 2 4
4 6 2 1 2 2 3 3 4 4 1 1 3 3 1	Yes 2 3 6 1 4 5

Зауваження

Відповідь для другого тесту зображена на малюнку нижче. Жовті ребра утворюють першу групу, а сині — другу.



Задача Н. Гра з одним переможцем

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1.5 seconds
Ліміт використання пам'яті: 256 megabytes

Задано неорієнтований граф з n вершинами та m ребрами. На кожному ребрі знаходиться певна кількість камінців. Також кожне ребро має певну вагу.

Два гравці грають в гру. Спочатку перший гравець обирає певну підмножину ребер S_1 з даного графу.

Далі другий гравець обирає непорожню підмножину ребер S_2 множини S_1 . Множина ребер S_2 повинна формувати набір реберно-простих циклів, що не перетинаються (тобто кожна компонента зв'язності повинна бути Ейлеровим циклом). Якщо такої підмножини S_2 не існує, другий гравець одразу програє.

Якщо другий гравець все ж таки зміг обрати підмножину ребер S_2 , то гравці починають грати в «Нім» з купками камінців, що знаходяться на ребрах з S_2 . На черговому ході гравець може забрати довільну додатну кількість камінців з будь-якої не пустої купки. Той, хто не може зробити хід, програє. Першим в грі «Нім» починає ходити перший гравець.

Знайдіть максимальну сумарну вагу підмножини ребер S_1 , яку може обрати перший гравець так, щоб він міг гарантовано виграти.

Формат вхідних даних

У першому рядку записано два цілі числа n, m ($2 \leq n \leq 64, 1 \leq m \leq 200\,000$) — кількість вершин та ребер початкового графу.

У наступних m рядках записано по чотири цілі числа u_i, v_i, a_i, w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 0 \leq a_i < 2^{60}, 1 \leq w_i \leq 10^9$) — номери вершин, що з'єднує відповідне ребро, кількість камінців та вагу ребра.

Формат вихідних даних

Виведіть єдине ціле число — максимальну сумарну вагу підмножини ребер S_1 , яку може обрати перший гравець так, щоб він міг гарантовано виграти.

Приклади

standard input	standard output
3 3 1 2 0 1 2 3 0 1 3 1 0 2	3
6 6 1 2 1 1 2 3 1 2 3 4 1 3 4 1 1 4 5 6 1 2 6 5 1 1	11
7 1 4 7 47 47	47

Зауваження

У першому прикладі перший гравець не може вибрати усі ребра, оскільки в такому випадку другий гравець їх і залишить (усі 3 ребра формують Ейлеров цикл). Подальшу гру в «Нім» перший гравець програє, оскільки не зможе зробити навіть перший хід.

Якщо ж перший гравець обере останні два ребра сумарною вагою 3, то другий не зможе сформувати з них Ейлеров цикл та одразу програє.

Задача I. Матроїд?

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	2 seconds
Ліміт використання пам'яті:	256 megabytes

Задано набір X , що складається з n предметів, пронумерованих від 0 до $n - 1$. Для кожної підмножини предметів $A \subset X$ відомо, чи є вона незалежною (відомо, чи виконується $A \in I$). Визначіть, чи формує пара $M = \langle X, I \rangle$ коректний матроїд.

Більш формально, необхідно перевірити, що множина I незалежних підмножин предметів задовольняє наступним аксіомам:

1. Пуста множина предметів є незалежною ($\emptyset \in I$).
2. Будь-яка підмножина незалежної множини також є незалежною (якщо $B \in I$ і $A \subset B$, то $A \in I$).
3. Якщо розмір незалежної множини A менше розміру незалежної множини B , то існує хоча б один елемент в B , який можна додати до A так, щоб множина A залишилася незалежною (якщо $A, B \in I$ і $|A| < |B|$, то $\exists x \in B \setminus A, A \cup \{x\} \in I$).

Формат вхідних даних

У першому рядку записано єдине ціле число n ($1 \leq n \leq 22$) — кількість предметів в наборі X .

У другому рядку знаходиться 2^n символів c_{mask} ($c_{mask} \in \{0, 1\}$), що описують матроїд. А саме: c_{mask} дорівнює 1 тоді і тільки тоді, коли відповідна множина предметів є незалежною ($A_{mask} \in I$). Нумерація предметів у бітовій масці починається з нуля, а біти записані в порядку від наймолодшого до найстаршого. Наприклад, бітова маска $mask = 5 = 101000 \dots 0_2$ позначає множину предметів $\{X_0, X_2\}$.

Формат вихідних даних

Якщо вхідні дані описують коректний матроїд, виведіть єдиний рядок «Yes».

Інакше спочатку виведіть єдиний рядок «No».

Далі якщо не виконується перша аксіома, то виведіть єдиний рядок «1».

Якщо не виконується друга аксіома, то виведіть єдиний рядок «2 A B», де A та B — дві бітові маски, для відповідних підмножин яких не виконується друга аксіома.

Якщо не виконується третя аксіома, то виведіть єдиний рядок «3 A B», де A та B — дві бітові маски, для відповідних підмножин яких не виконується третя аксіома.

Якщо не виконуються одразу декілька аксіом, ви можете вивести будь-яку з них.

Бітові маски необхідно виводити як послідовність символів f_0, f_1, \dots, f_{n-1} ($f_i \in \{0, 1\}$), що позначають присутність відповідного предмету в множині.

Приклади

standard input	standard output
3 11101110	Yes
4 0111111111111111	No 1
2 1001	No 2 01 11
3 11111000	No 3 001 110

Задача J. Перетин мартоїдів

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Мартоїдом називається бінарна матриця розміру $n \times m$, у кожній клітинці якої записано «0» або «1». Для того щоб матриця була коректним мартоїдом, для неї повинні виконуватись дві додаткові умови:

- Існує хоча б одна клітинка, у якій записано символ «1».
- Між кожною парою «одиничних» клітинок існує шлях по «одиничним» клітинкам між ними, усі сусідні клітинки якого мають спільну сторону.

Перетином двох мартоїдів A та B називається матриця C розміру $n \times m$, у кожній клітинці якої записано «1» тоді і тільки тоді, коли обидва відповідні значення цих двох мартоїдів також дорівнюють «1». Зверніть увагу, що перетин двох мартоїдів не обов'язково є коректним мартоїдом.

Задано бінарну матрицю C . Гарантується, що її **крайні елементи дорівнюють нулю**, тобто $C_{i,j} = 0$ при $i = 1, i = n, j = 1, j = m$. Знайдіть два мартоїди A та B , перетин яких дорівнює C .

Формат вхідних даних

У першому рядку записано два цілі числа n, m ($3 \leq n, m \leq 500$).

Наступні n рядків містять по m символів $C_{i,1}, C_{i,2}, \dots, C_{i,m}$ ($C_{i,j} \in \{«0», «1»\}$).

Формат вихідних даних

Якщо шуканих мартоїдів не існує, виведіть єдиний рядок «Impossible».

Інакше виведіть два мартоїди A та B , перетин яких дорівнює заданій матриці.

Перші n рядків повинні описувати мартоїд A у форматі, аналогічному формату вхідних даних.

Далі повинен бути виведений пустий рядок.

Останні n рядків повинні описувати мартоїд B у форматі, аналогічному формату вхідних даних.

Якщо існує кілька правильних відповідей, дозволяється вивести будь-яку з них.

Приклади

standard input	standard output
5 5 00000 01010 00000 01010 00000	00000 11111 10001 11111 00000 01110 01010 01010 01010 01110
3 7 0000000 0110110 0000000	0111110 0110110 0111110 0000000 1111111 0000000

Задача К. Унікальні суми

Назва вхідного файлу: `standard input`
Назва вихідного файлу: `standard output`
Ліміт часу: 1.5 seconds
Ліміт використання пам'яті: 256 megabytes

Задано масив цілих чисел a з n елементів, кожен з яких дорівнює -1 , 0 , або 1 . Знайдіть кількість різних значень, які може приймати вираз $\sum_{i=l}^r a_i$, де $1 \leq l \leq r \leq n$.

Формат вхідних даних

У першому рядку записано єдине ціле число n ($1 \leq n \leq 10^6$) — довжина масиву.

У другому рядку записано n цілих чисел a_1, a_2, \dots, a_n ($-1 \leq a_i \leq 1$).

Формат вихідних даних

Виведіть єдине ціле число — відповідь на задачу.

Приклад

standard input	standard output
5 1 -1 1 0 -1	3

Задача L. Якщо вам стало нудно

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	15 seconds
Ліміт використання пам'яті:	256 megabytes

Коли Мартоїду становиться нудно, він починає грати зі своєю улюбленою іграшкою — електронним дісплеєм. На цьому дісплеї відображуються п'ять цифр, а тому він може показувати будь-яке п'ятизначне число від 0 до 99999. Спочатку на дісплеї відображується число a ($0 \leq a \leq 99999$).

Гра з дісплеєм полягає в наступному. Дісплей обирає два числа L, R ($1 \leq L < R \leq 99999$) та повідомляє їх Мартоїду. Після цього дісплей загадує число x на інтервалі $[L; R]$. Мартоїду необхідно його відгадати. Для цього він може виконувати наступні операції:

- змінити значення будь-якої цифри дісплею;
- запитати результат порівняння загаданого числа x та числа y , що на поточний момент відображається на дісплеї: чи загадане число x менше, рівне, або більше числа y .

Знайдіть мінімальну кількість операцій, за яку Мартоїд гарантовано може відгадати число x .

Формат вхідних даних

У першому рядку записано єдине ціле число t ($1 \leq t \leq 50$) — кількість тестів.

У кожному з наступних t рядків знаходяться по три цілі числа a, L, R ($0 \leq a \leq 99999$, $1 \leq L < R \leq 99999$) — початкове число, що відображується на дісплеї, та інтервал, на якому загадано число x .

Формат вихідних даних

Для кожного з тестів виведіть єдине ціле число — мінімальну кількість операцій, за яку Мартоїд гарантовано можна відгадати число x .

Приклад

standard input	standard output
4	1
47 46 48	6
0 97 107	4
77744 12043 12045	8
100 61 69	

Зауваження

У першому тесті можна одразу виконати операцію другого типу. При результаті «менше» $x = 46$, при результаті «більше» $x = 48$, інакше $x = 47$. Саме тому однієї операції достатньо, щоб гарантовано відгадати число x .