

UCPC 2026 - I stage

Ukrainian Collegiate Programming Contest

3 жовтня 2026 р.

Задачі

A Антивірус
Час 1,5 с · Пам’ять 256 МіБ

B Колекційні пари
Час 0,5 с · Пам’ять 256 МіБ

C Танок залишків
Час 1 с · Пам’ять 256 МіБ

D Парні та непарні
Час 1,5 с · Пам’ять 256 МіБ

E Чотири числа
Час 2 с · Пам’ять 256 МіБ

F Переплутані покази
Час 3 с · Пам’ять 256 МіБ

G Числова гра
Час 1 с · Пам’ять 256 МіБ

H Візерунок
Час 1 с · Пам’ять 256 МіБ

I Круглий стіл
Час 2 с · Пам’ять 512 МіБ

J Терези
Час 0,5 с · Пам’ять 256 МіБ · Інтерактивна

K Сума і добуток
Час 1 с · Пам’ять 256 МіБ

L Три комп'ютери
Час 3 с · Пам’ять 256 МіБ

M Гірлянда UCPC
Час 4,5 с · Пам’ять 256 МіБ

N Щоденник погоди
Час 0,5 с · Пам’ять 256 МіБ

A Антивірус

Ліміт часу 1,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

На комп'ютері Остапа є n папок, пронумерованих від 1 до n. Папка 1 — коренева, а кожна інша папка i $(2 \le i \le n)$ лежить безпосередньо всередині папки a_i; гарантується, що $a_i < i$.

Остап k разів запускав антивірус. j-й запуск перевіряє папку c_j і всі папки всередині неї, що лежать не глибше ніж на d_j рівнів від c_j.

Формально, папка v перевірена j-м запуском, якщо з c_j можна дістатися до v, переходячи від папки до папки, що лежить безпосередньо в ній, не більше ніж d_j разів. Зокрема, папка c_j завжди перевірена j-м запуском, навіть коли $d_j = 0$.

Одну й ту саму папку могли перевіряти кілька разів. Знайдіть кількість папок, перевірених принаймні одним запуском.

Вхідні дані

Перший рядок містить два цілі числа n і k $(2 \le n \le 3 \cdot 10^5,\ 1 \le k \le 3 \cdot 10^5)$ — кількість папок і кількість запусків антивірусу.

Другий рядок містить $n-1$ цілих чисел $a_2, a_3, \ldots, a_n$ $(1 \le a_i < i)$, де a_i — папка, у якій створено папку i.

Кожен з наступних k рядків містить два цілі числа c_j і d_j $(1 \le c_j \le n,\ 0 \le d_j \le 3 \cdot 10^5)$ — папку та глибину j-го запуску.

Вихідні дані

Виведіть одне ціле число — кількість папок, перевірених принаймні одним запуском.

Приклади

Вхідні дані

```
8 3
1 1 2 2 3 5 5
2 1
3 0
8 2
```

Вихідні дані

```
5
```

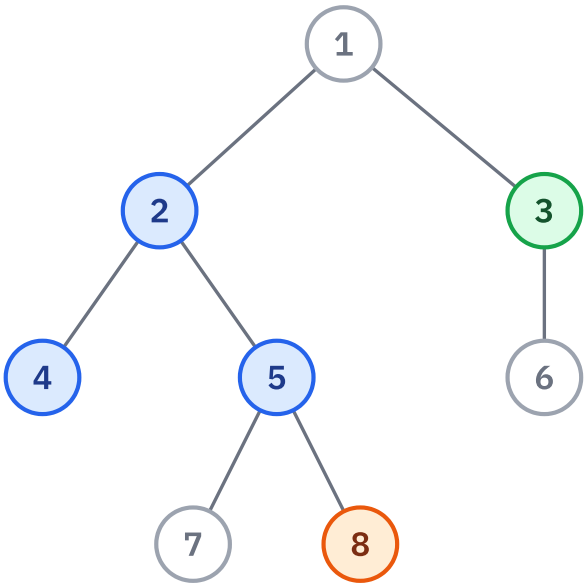
Вхідні дані

```
6 4
1 2 2 4 1
2 2
2 0
1 0
4 0
```

Вихідні дані

```
5
```

Примітка



У першому прикладі папка 1 містить папки 2 і 3, папка 2 — папки 4 і 5, папка 3 — папку 6, а папка 5 — папки 7 і 8. Перший запуск перевіряє папки 2, 4 і 5 (синій колір); другий — лише папку 3 (зелений колір); третій — лише папку 8 (червоний колір), бо в ній немає інших папок. Усього перевірено 5 папок.

У другому прикладі перший запуск перевіряє папки 2, 3, 4 і 5, другий — папку 2, третій — папку 1, а четвертий — папку 4. Папка 6 не перевірена, тож відповідь дорівнює 5.

B Колекційні пари

Ліміт часу 0,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Марта колекціонує банкноти й оцінює їх за серійними номерами. Серійний номер — це додатне ціле число, і Марта називає його *красивим*, якщо воно ділиться на суму своїх цифр у десятковому записі. Наприклад, 110 красиве, бо сума його цифр рівна $1+1+0=2$ і $110=2\cdot 55$; число 13 не красиве, бо 13 не ділиться на $1+3=4$.

Найбільше Марта цінує *красиві пари*: дві банкноти, серійні номери яких — це два послідовні красиві цілі числа x і $x+1$. Наприклад, 110 і 111 утворюють красиву пару, адже 111 ділиться на $1+1+1=3$.

Марта збирається відвідати q колекціонерів. У i-го колекціонера є пачка, що містить рівно по одній банкноті з кожним серійним номером від m_i до $2m_i$ включно. Для кожного колекціонера вона хоче вибрати красиву пару в його пачці, тобто два послідовні **красиві** цілі числа x і $x+1$, для яких

$$m_i \le x \quad \text{і} \quad x+1 \le 2m_i,$$

або визначити, що такої пари немає.

Вхідні дані

Перший рядок містить одне ціле число q ($1 \le q \le 10^5$) — кількість колекціонерів.

Кожен із наступних q рядків містить одне ціле число m_i ($m_i \ge 1$), записане в десятковій системі без провідних нулів. Сумарна кількість цифр у всіх m_i не перевищує 10^6 ($\sum |m_i| \le 10^6$).

Вихідні дані

Виведіть q рядків, i-й з яких є відповіддю для i-го колекціонера.

Якщо в пачці i-го колекціонера є красива пара $(x, x+1)$, виведіть число x у десятковому записі без провідних нулів. Якщо можливих відповідей декілька, виведіть будь-яку з них.

Якщо в пачці немає красивої пари, виведіть одне число -1.

Приклади

Вхідні дані	Вихідні дані
5 1 8 30 2718 987654321987654321	1 8 -1 2754 987654321987655230

Примітка

У першого колекціонера в пачці є банкноти 1 і 2; обидва числа красиві, тож вони утворюють єдину можливу красиву пару.

У другого колекціонера серійні номери — $8, 9, \ldots, 16$. Серед них красивими є 8, 9, 10 і 12, тож можливі рівно дві красиві пари: пари $(8, 9)$ і $(9, 10)$.

У третього колекціонера серійні номери — $30, 31, \ldots, 60$; жодні два послідовні числа з них не є одночасно красивими.

Для четвертого колекціонера одна з можливих відповідей — це $2754 = 18 \cdot 153$ і $2755 = 19 \cdot 145$, де 18 і 19 — їх суми цифр.

Для п'ятого колекціонера суми цифр чисел 987654321987655230 і 987654321987655231 дорівнюють 90 і 91, і виконується, що $987654321987655230 = 90 \cdot 10973936910973947$ та $987654321987655231 = 91 \cdot 10853344197666541$.

C Танок залишків

Ліміт часу 1 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Троє гномів — Мох, Лист і Корінь — танцюють по колу навколо дуба: за Мохом іде Лист, за Листом — Корінь, а за Коренем — Мох.

Перед танцем кожен гном набирає собі жолудів: Мох бере a жолудів, де $1 \le a \le n$; Лист бере b жолудів, де $1 \le b \le m$; Корінь бере c жолудів, де $1 \le c \le k$.

Кожен гном розкладає свої жолуді на якомога більше купок, у кожній з яких стільки жолудів, скільки їх має наступний гном. Жолуді, що не потрапили в купки, — це *залишок* гнома. Отже, залишки Моха, Листа і Кореня дорівнюють $a \bmod b$, $b \bmod c$ і $c \bmod a$, де $x \bmod y$ — остача від ділення x на y (наприклад, $7 \bmod 3 = 1$, $4 \bmod 11 = 4$).

Танець називається *гармонійним*, якщо залишки всіх трьох гномів однакові:

$$a \bmod b = b \bmod c = c \bmod a.$$

Для кожного з t тестових випадків знайдіть кількість трійок (a, b, c), для яких танець гармонійний.

Вхідні дані

Перший рядок містить одне ціле число t ($1 \le t \le 10^5$) — кількість тестових випадків.

Кожен з наступних t рядків містить три цілі числа n, m і k ($1 \le n, m, k \le 10^9$).

Вихідні дані

Для кожного тестового випадку виведіть в окремому рядку одне ціле число — кількість трійок (a, b, c), для яких танець гармонійний.

Приклади

Вхідні дані	Вихідні дані
2 2 3 1 3 8 5	1 3

Примітка

У першому тестовому випадку Корінь може взяти лише один жолудь. Єдина підхожа трійка — $(1, 1, 1)$: усі три залишки дорівнюють 0. Наприклад, трійка $(2, 1, 1)$ не підходить: залишки Моха і Листа дорівнюють 0, а залишок Кореня — $1 \bmod 2 = 1$.

D Парні та непарні

Ліміт часу 1,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Нехай A — натуральне число, а $a_{m-1}a_{m-2}\ldots a_1a_0$ — його десятковий запис без провідних нулів, тобто

$$A = a_0 + 10 \cdot a_1 + 10^2 \cdot a_2 + \ldots + 10^{m-1} \cdot a_{m-1}.$$

Іншими словами, цифри нумеруються **справа наліво, починаючи з нуля**: a_0 — остання цифра (цифра одиниць), a_1 — цифра десятків і так далі. Число i називається *позицією* цифри a_i.

Число A називається *збалансованим*, якщо

$$S_{\text{odd}}(A) - P_{\text{odd}}(A) = S_{\text{even}}(A) - P_{\text{even}}(A).$$

Тут:

- $S_{\text{odd}}(A)$ — сума всіх цифр a_i, які є непарними числами;
- $S_{\text{even}}(A)$ — сума всіх цифр a_i, які є парними числами;
- $P_{\text{odd}}(A)$ — сума всіх цифр a_i, що стоять на непарних позиціях i;
- $P_{\text{even}}(A)$ — сума всіх цифр a_i, що стоять на парних позиціях i.

Наприклад, для $A = 1234$ маємо $a_0 = 4$, $a_1 = 3$, $a_2 = 2$, $a_3 = 1$. Тоді:

- $S_{\text{odd}} = 1 + 3 = 4$;
- $S_{\text{even}} = 2 + 4 = 6$;
- $P_{\text{odd}} = 3 + 1 = 4$;
- $P_{\text{even}} = 4 + 2 = 6$.

Оскільки виконується $S_{\text{odd}} - P_{\text{odd}} = 4 - 4 = 0$ і $S_{\text{even}} - P_{\text{even}} = 6 - 6 = 0$, число 1234 збалансоване.

Проте для $A = 12340$ позиції всіх початкових цифр збільшуються на одиницю: $a_0 = 0$, $a_1 = 4$, $a_2 = 3$, $a_3 = 2$, $a_4 = 1$. Тепер:

- $S_{\text{odd}} = 4$;
- $S_{\text{even}} = 6$;
- $P_{\text{odd}} = 4 + 2 = 6$;
- $P_{\text{even}} = 0 + 3 + 1 = 4$.

Оскільки $4 - 6 \neq 6 - 4$, число 12340 не є збалансованим.

Задано Q запитів. У кожному запиті дано натуральне число N. Знайдіть **найменше** збалансоване число M, десятковий запис якого *починається з* десяткового запису числа N. Іншими словами, M отримується дописуванням праворуч від N ще $k \geq 0$ цифр (будь-яких, зокрема нулів); випадок $k = 0$, тобто $M = N$, дозволений. Можна довести, що таке число M завжди існує.

Вхідні дані

Перший рядок містить одне ціле число Q ($1 \leq Q \leq 10^6$) — кількість запитів.

Кожен з наступних Q рядків містить одне натуральне число N, записане без провідних нулів. Гарантується, що $1 \leq |N| \leq 10^6$ та сума $|N|$ за всіма запитами не перевищує 10^6, де $|N|$ — це кількість цифр числа N.

Вихідні дані

Для кожного запиту виведіть шукане число M в окремому рядку повністю, без провідних нулів.

Приклади

Вхідні дані
```
5
1234
9
1
99
98
```

Вихідні дані
```
1234
90
10
992081
98
```

Вхідні дані
```
3
88
2025
999999999
```

Вихідні дані
```
88107
20250
99999999904080808080
```

Примітка

У першому прикладі:

- 1234 саме є збалансованим (див. вище), тож $M = N$.
- 9 не є збалансованим: $S_{\text{odd}} - P_{\text{odd}} = 9 - 0 = 9$, тоді як $S_{\text{even}} - P_{\text{even}} = 0 - 9 = -9$. Наступні кандидати — це $90, 91, \ldots, 99$; серед них 90 збалансоване.
- Для 99 відповідь дорівнює 992081: тут $a_0 = 1$, $a_1 = 8$, $a_2 = 0$, $a_3 = 2$, $a_4 = 9$, $a_5 = 9$, тож $S_{\text{odd}} - P_{\text{odd}} = (1 + 9 + 9) - (8 + 2 + 9) = 0$ і $S_{\text{even}} - P_{\text{even}} = (8 + 0 + 2) - (1 + 0 + 9) = 0$. Жодне менше число не підходить.

E Чотири числа

Ліміт часу 2 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Оленка має послідовність із n цілих чисел $a_1, a_2, \ldots, a_n$ та ціле число m.

Вона хоче вибрати чотири індекси i, j, k, t ($1 \le i, j, k, t \le n$) такі, що $a_i + a_j + a_k + a_t \le m$. Зверніть увагу, що індекси **не обов'язково** різні (деякі з них можуть збігатися).

Серед усіх таких четвірок Оленку цікавить найбільше можливе значення $a_i + a_j + a_k + a_t$. Допоможіть їй знайти його або повідомте, що жодної такої четвірки індексів не існує.

Вхідні дані

Перший рядок містить два цілі числа n та m ($1 \le n \le 3000$, $-4 \cdot 10^{18} \le m \le 4 \cdot 10^{18}$).

Другий рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($-10^{18} \le a_i \le 10^{18}$).

Вихідні дані

Виведіть одне ціле число — максимальне значення $a_i + a_j + a_k + a_t$ серед усіх четвірок індексів, для яких $a_i + a_j + a_k + a_t \le m$.

Якщо такої четвірки не існує, виведіть число -1.

Приклади

Вхідні дані
```
4 20
3 9 1 7
```
Вихідні дані
```
20
```

Вхідні дані
```
3 -10
5 -2 8
```
Вихідні дані
```
-1
```

Вхідні дані
```
5 2
-5 4 10 -1 6
```
Вихідні дані
```
2
```

Примітка

У першому прикладі можна взяти $i = j = 2$ та $k = t = 3$: $a_2 + a_2 + a_3 + a_3 = 9 + 9 + 1 + 1 = 20 \le 20$.

У другому прикладі навіть найменша можлива сума $a_2 + a_2 + a_2 + a_2 = -8$ більша за $m = -10$, тож відповіді не існує.

У третьому прикладі одним з оптимальних виборів є $i = 1$, $j = k = 2$, $t = 4$: $(-5) + 4 + 4 + (-1) = 2$.

F Переплутані покази

Ліміт часу 3 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Старий алхімік Орест зберігає колекцію з n магічних кристалів; i-й кристал має додатний цілий *заряд* a_i. Заряди різних кристалів можуть збігатися.

В Ореста є два прилади. Якщо покласти непорожню групу кристалів на *терези*, вони покажуть сумарний заряд групи, тобто **суму** зарядів. Якщо ж ту саму групу покласти на *резонатор*, він покаже **побітове АБО** (OR) їхніх зарядів.

Однієї ночі Орест виміряв кожну з $2^n - 1$ непорожніх груп кристалів на обох приладах і записав усі покази до журналу. Отже, журнал містить рівно $2 \cdot (2^n - 1)$ чисел. На жаль, записи були на окремих аркушах, які перемішалися: тепер невідомо, якій групі належить кожен показ і який прилад його видав. Самі кристали вже продано.

За $2 \cdot (2^n - 1)$ показами в довільному порядку відновіть заряди кристалів.

Формально, для масиву $a_1, a_2, \ldots, a_n$ і непорожньої множини індексів $S \subseteq \{1, \ldots, n\}$ позначимо $\mathrm{sum}(S) = \sum_{i \in S} a_i$ та $\mathrm{or}(S) = a_{i_1} \mid a_{i_2} \mid \ldots \mid a_{i_k}$, де $S = \{i_1, \ldots, i_k\}$, а $\mid$ — побітове АБО. Журнал — це мультимножина, що складається з чисел $\mathrm{sum}(S)$ та $\mathrm{or}(S)$ для всіх $2^n - 1$ непорожніх множин S (якщо $\mathrm{sum}(S) = \mathrm{or}(S)$, то для цієї S таке значення входить двічі).

Вхідні дані

У першому рядку задано одне ціле число t ($1 \le t \le 10^5$) — кількість тестових випадків.

Кожен тестовий випадок складається з двох рядків. У першому рядку задано ціле число n ($1 \le n \le 20$) — кількість кристалів. У другому рядку задано $2 \cdot (2^n - 1)$ цілих чисел $b_1, b_2, \ldots, b_{2(2^n-1)}$ ($1 \le b_j \le 2 \cdot 10^{10}$) — покази з журналу в довільному порядку.

Гарантується, що покази кожного тестового випадку є в точності журналом деякого масиву з n цілих чисел $a_1, \ldots, a_n$, де $1 \le a_i \le 10^9$.

Сума 2^n за всіма тестовими випадками не перевищує 2^{20}.

Вихідні дані

Для кожного тестового випадку виведіть в окремому рядку n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) у довільному порядку такі, що журнал цього масиву збігається з заданими показами як мультимножина. Якщо відповідей кілька, виведіть будь-яку.

Приклади

Вхідні дані

```
4
1
7 7
2
3 1 2 3 2 1
2
2 4 2 2 2 2
3
7 5 3 14 6 7 11 5 8 7 3 9 6 7
```

Вихідні дані

```
7
1 2
2 2
3 5 6
```

Примітка

У першому тестовому випадку єдиний кристал має заряд 7: обидва прилади показують 7.

У другому тестовому випадку заряди дорівнюють 1 і 2: терези показують 1, 2, 3, а резонатор — 1, 2, 3.

У третьому тестовому випадку обидва кристали мають заряд 2: терези показують 2, 2, 4, а резонатор — 2, 2, 2.

У четвертому тестовому випадку заряди дорівнюють 3, 5, 6. Терези показують 3, 5, 6, 8, 9, 11, 14, а резонатор — 3, 5, 6, 7, 7, 7, 7.

G Числова гра

Ліміт часу 1 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Марта грає в гру з числом, записаним на дошці у двійковій системі числення без провідних нулів. Спочатку на дошці записано додатне число n.

За один *хід* Марта виконує дві дії в такому порядку:

1. Стирає з запису будь-яку кількість нулів (можливо, жодного) і дописує будь-яку кількість нулів (можливо, жодного) у будь-які місця запису так, щоб він не починався з нуля. Одиниці стирати чи дописувати **не можна**.
2. Збільшує записане число на 1 і записує результат замість нього, знову у двійковій системі числення без провідних нулів.

Марта виграє, якщо після одного з її ходів на дошці буде записано додатне число m. Якщо $n = m$, вона виграє одразу, не зробивши жодного ходу. Знайдіть найменшу кількість ходів, за яку Марта може виграти, або повідомте, що виграти неможливо.

Вхідні дані

Перший рядок містить одне ціле число t ($1 \le t \le 1000$) — кількість тестових випадків.

Кожен з наступних t рядків містить два додатні числа n і m ($1 \le n, m$), записані у двійковій системі числення без провідних нулів і розділені пробілом.

Гарантується, що $1 \le |n|, |m| \le 1000$, де $|x|$ — кількість цифр у двійковому записі числа x.

Вихідні дані

Для кожного тестового випадку виведіть в окремому рядку одне ціле число — найменшу кількість ходів, або -1, якщо виграти неможливо.

Приклади

Вхідні дані

```
6
110 1110
1 10000
111 1
101 101
1000 11
11111 1001
```

Вихідні дані

```
2
4
-1
0
1
2
```

Примітка

У першому тестовому випадку $n = 6 =$ `110`, $m = 14 =$ `1110`. Марта може виграти за два ходи:

- Хід 1: Марта дописує нуль у кінець, `110` $\to$ `1100` $= 12$, і збільшує число: $13 =$ `1101`.
- Хід 2: Марта нічого не стирає й не дописує і збільшує число: $14 =$ `1110`.

У третьому тестовому випадку $m = 1$ отримати неможливо.

У четвертому тестовому випадку $n = m$, тож Марта виграє одразу.

H Візерунок

Ліміт часу 1 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Оленка робить штамп для візерунка на стрічці. Штамп — це смужка з n клітинок, кожну з яких пофарбовано в один з k кольорів. Позначимо кольори клітинок зліва направо $w_1, w_2, \ldots, w_n$ $(1 \le w_i \le k)$.

Щоб відбитки на стрічці не можна було сплутати, Оленці потрібен вдалий штамп. Штамп називається *вдалим*, якщо жоден його початок не пофарбований так само, як кінець такої самої довжини. Тобто для кожного l від 1 до $n-1$ перші l клітинок відрізняються від останніх l:

$$w_1 w_2 \ldots w_l \neq w_{n-l+1} w_{n-l+2} \ldots w_n.$$

Штамп з однієї клітинки завжди вдалий.

Наприклад, при $k = 3$ штампи `123`, `112` і `3` вдалі, а штампи `121` $(l = 1)$, `12312` $(l = 2)$ і `1111` $(l = 1, 2, 3)$ — ні.

Задано t запитів. Для кожного запиту порахуйте кількість вдалих штампів з n клітинок, пофарбованих у k кольорів. Оскільки відповіді можуть бути дуже великими, виведіть їх за модулем $10^9 + 7$.

Вхідні дані

Перший рядок містить одне ціле число t $(1 \le t \le 10^5)$ — кількість запитів.

Кожен із наступних t рядків містить два цілі числа n і k $(1 \le n \le 10^6,\ 1 \le k \le 10^9)$ — кількість клітинок штампа і кількість кольорів.

Гарантується, що сума n по всіх запитах не перевищує 10^6.

Вихідні дані

Для кожного запиту виведіть в окремому рядку кількість вдалих штампів з n клітинок, пофарбованих у k кольорів, за модулем $10^9 + 7$.

Приклади

Вхідні дані

```
4
1 5
2 3
3 2
4 2
```

Вихідні дані

```
5
6
4
6
```

Вхідні дані

```
2
5 3
30 1000000000
```

Вихідні дані

```
144
508301895
```

Примітка

У першому прикладі:

- Для $n = 1$ усі 5 штампів вдалі.

- Для $n = 2$, $k = 3$ штамп вдалий тоді й лише тоді, коли його дві клітинки мають різні кольори: $3 \cdot 2 = 6$ штампів.
- Для $n = 3$, $k = 2$ вдалими є штампи `112`, `122`, `211` і `221`.
- Для $n = 4$, $k = 2$ вдалими є штампи `1112`, `1122`, `1222`, `2111`, `2211` і `2221`. Наприклад, штамп `1212` не є вдалим, бо його перші дві клітинки `12` пофарбовано так само, як дві останні, а штамп `1221` не є вдалим, бо він починається й закінчується кольором 1.

I Круглий стіл

Ліміт часу 2 с · Ліміт пам'яті 512 МіБ · Стандартний ввід/вивід

За круглим столом сидять n студентів гуртка з програмування. Вони пронумеровані від 1 до n за годинниковою стрілкою, тож студент n сидить поруч зі студентом 1.

Тренер хоче сформувати з них якомога більше команд. Щоб члени команди могли обговорювати задачі, не перекрикуючи одне одного через увесь стіл, він склав список з m можливих команд, кожна з яких складається зі студентів, що сидять підряд. Команди можуть мати будь-який розмір.

i-ту можливу команду задано числами l_i і r_i: до неї входять студенти з номерами від l_i до r_i включно, якщо рухатися за годинниковою стрілкою. Тобто:

- якщо $l_i \le r_i$, команда складається зі студентів $l_i, l_i + 1, \dots, r_i$;
- якщо $l_i > r_i$, команда складається зі студентів $l_i, l_i + 1, \dots, n, 1, 2, \dots, r_i$.

Зокрема, при $l_i = r_i$ команда складається з одного студента, а при $l_i = r_i + 1$ або $l_i = 1$ і $r_i = n$ — з усіх n студентів. Деякі команди в списку можуть збігатися.

Кожен студент може входити щонайбільше до **однієї** сформованої команди. Знайдіть найбільшу кількість команд, які можна сформувати.

Вхідні дані

Перший рядок містить одне ціле число t ($1 \le t \le 10^6$) — кількість тестових випадків.

Перший рядок кожного тестового випадку містить два цілі числа n і m ($1 \le n, m \le 10^6$) — кількість студентів і кількість можливих команд.

Кожен із наступних m рядків містить два цілі числа l_i і r_i ($1 \le l_i, r_i \le n$) — i-ту можливу команду.

Гарантується, що сума n і сума m по всіх тестових випадках не перевищують 10^6.

Вихідні дані

Для кожного тестового випадку виведіть в окремому рядку одне ціле число — найбільшу кількість команд, які можна сформувати одночасно.

Приклади

Вхідні дані

```
3
5 3
5 1
2 3
3 4
4 2
2 1
1 4
6 5
1 3
3 4
5 6
6 2
4 4
```

Вихідні дані

```
2
1
3
```

Вхідні дані

```
1
10 6
9 2
3 5
6 8
1 4
5 7
4 4
```

Вихідні дані

```
3
```

Примітка

У першому прикладі:

- У першому тестовому випадку можливі команди — $\{5, 1\}$, $\{2, 3\}$ і $\{3, 4\}$. Команди $\{5, 1\}$ і $\{2, 3\}$ можна сформувати разом, а всі три — не можна, бо дві останні мають спільного студента 3.
- У другому тестовому випадку обидві команди складаються з усіх чотирьох студентів, тож сформувати можна лише одну.
- У третьому тестовому випадку можна сформувати $\{1, 2, 3\}$, $\{4\}$ і $\{5, 6\}$. Більше сформувати неможливо.

У другому прикладі можливі команди — $\{9, 10, 1, 2\}$, $\{3, 4, 5\}$, $\{6, 7, 8\}$, $\{1, 2, 3, 4\}$, $\{5, 6, 7\}$ і $\{4\}$. Перші три з них можна сформувати разом, а чотири — не можна.

J Терези

Ліміт часу 0,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід · Інтерактивна задача

У журі є старі терези і два секретні цілі числа a та b, де $0 \le a, b < 2^n$, а число n вам відоме. Ваше завдання — знайти обидва секретні числа.

Терези трохи незвичайні. Ви можете дати журі два цілі числа x та y ($0 \le x, y < 2^n$). Тоді журі кладе на ліву шальку число $a \oplus x$, а на праву — число $b \oplus y$, і повідомляє, яка шалька важча, або що терези врівноважені. Іншими словами, ви дізнаєтеся результат порівняння $a \oplus x$ з $b \oplus y$. Тут $\oplus$ позначає побітове виключне АБО (XOR).

Скористатися терезами можна не більше ніж $2n$ разів.

Вхідні дані

Перший рядок містить одне ціле число n ($2 \le n \le 60$).

Взаємодія

Щоб виконати зважування, виведіть рядок `? x y` з двома цілими числами x та y ($0 \le x, y < 2^n$). У відповідь журі виведе рядок з одним символом:

- `<`, якщо $a \oplus x < b \oplus y$;
- `=`, якщо $a \oplus x = b \oplus y$;
- `>`, якщо $a \oplus x > b \oplus y$.

Коли ви визначите секретні числа, виведіть рядок `! a b` з числами a та b і завершіть роботу програми. Цей рядок не вважається зважуванням.

Після виведення кожного рядка не забудьте вивести символ кінця рядка та скинути буфер виводу (наприклад, `fflush(stdout)` або `cout.flush()` у C++, `sys.stdout.flush()` у Python, `System.out.flush()` у Java). Інакше ви можете отримати вердикт «Перевищено ліміт часу». Якщо ви виконаєте більше ніж $2n$ зважувань, або зважування з числами поза допустимими межами, або виведете рядок у неправильному форматі, ви отримаєте вердикт «Неправильна відповідь».

Журі може бути **адаптивним**: числа a та b не зафіксовані заздалегідь і можуть залежати від ваших зважувань. Проте в кожен момент існує принаймні одна пара (a, b), яка узгоджується з усіма наданими відповідями. Тому виводьте відповідь лише тоді, коли пара визначена однозначно: якщо з усіма відповідями узгоджується ще якась інша пара, журі може вважати секретною саме її, і вашу відповідь буде визнано неправильною.

Приклади

Вхідні дані

```
3

>

<

<

=

>
```

Вихідні дані

```
? 0 0

? 4 0

? 0 4

? 4 2

? 4 3

! 5 3
```

Вхідні дані

```
2

<

=

>
```

Вихідні дані

```
? 0 0

? 0 2

? 0 3

! 1 3
```

Примітка

У прикладах у стовпчику вхідних даних наведено рядки, які виводить журі, а у стовпчику вихідних даних — рядки, які виводить ваша програма, у тому порядку, в якому відбувається обмін. Порожні рядки додано лише для вирівнювання стовпчиків; вони не є частиною взаємодії.

У першому прикладі $n = 3$, а секретні числа дорівнюють $a = 5$ та $b = 3$.

- `? 0 0` порівнює $a = 5$ з $b = 3$, відповідь `>`;
- `? 4 0` порівнює $5 \oplus 4 = 1$ з 3, відповідь `<`;
- `? 0 4` порівнює 5 з $3 \oplus 4 = 7$, відповідь `<`;
- `? 4 2` порівнює $5 \oplus 4$ з $3 \oplus 2$, тобто 1 з 1, відповідь `=`;
- `? 4 3` порівнює $5 \oplus 4$ з $3 \oplus 3$, тобто 1 з 0, відповідь `>`.

Щоб надати відповідь, програма виводить `! 5 3`. Використано п'ять зважувань, тобто в межах ліміту $2n$.

У другому прикладі $n = 2$, $a = 1$, $b = 3$.

- `? 0 0`, відповідь `<`;
- `? 0 2` порівнює a з $b \oplus 2$, відповідь `=`;
- `? 0 3` порівнює a з $b \oplus 3$, відповідь `>`.

Використано три зважування, тобто в межах ліміту $2n$.

K Сума і добуток

Ліміт часу 1 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Оленка щойно дізналася про дійсні числа й одразу вигадала гру. Вона називає дійсне число s, а її старший брат має записати на дошці одне або кілька дійсних чисел $a_1, a_2, \dots, a_n$ так, щоб їхня сума дорівнювала s, а добуток дорівнював 1:

$$a_1 + a_2 + \dots + a_n = s, \qquad a_1 \cdot a_2 \cdot \dots \cdot a_n = 1.$$

Брат лінивий, тому хоче записати якомога менше чисел. Допоможіть йому: для кожного числа, яке назвала Оленка, знайдіть найменше можливе $n \geq 1$ і будь-які відповідні числа $a_1, \dots, a_n$.

Вхідні дані

Перший рядок містить одне ціле число t ($1 \leq t \leq 10^4$) — кількість чисел, які назвала Оленка.

Кожен із наступних t рядків містить одне дійсне число s ($|s| \leq 10^9$), записане рівно з трьома цифрами після десяткової крапки.

Вихідні дані

Для кожного з t чисел виведіть два рядки. У першому рядку має бути ціле число n — найменша кількість дійсних чисел, сума яких **точно** дорівнює s, а добуток **точно** дорівнює 1. У другому рядку мають бути n дійсних чисел $a_1, a_2, \dots, a_n$, розділених пробілами. Можна довести, що такі числа завжди існують.

Оскільки числа виводяться з обмеженою точністю, відповідь зараховується, якщо n найменше можливе, а виведені числа задовольняють такі умови:

- $|a_i| \leq 10^{10}$ для кожного i;
- $|(a_1 + a_2 + \dots + a_n) - s| \leq 10^{-6} \cdot \max(1, |s|)$;
- $|a_1 \cdot a_2 \cdot \dots \cdot a_n - 1| \leq 10^{-6}$.

Числа можна виводити як у звичайному десятковому записі (наприклад, `-0.618034`), так і в експоненційному (наприклад, `1.5e-07`).

Приклади

Вхідні дані

```
5
1.000
2.500
0.500
-2.000
0.000
```

Вихідні дані

```
1
1
2
0.5 2
3
-1 2 -0.5
2
-1 -1
3
-1 1.6180339887 -0.6180339887
```

Примітка

У прикладі для $s = 1$ достатньо одного числа. Для $s = 2.5$ можна взяти $0.5 + 2 = 2.5$ і $0.5 \cdot 2 = 1$. Для $s = 0.5$ не підходить ні одне число, ні пара чисел, а трійка $-1, 2, -0.5$ підходить.

Для $s = 0$ виведені числа -1, 1.6180339887, -0.6180339887 правильні лише наближено (точні значення — $\frac{1\pm\sqrt{5}}{2}$), але вони задовольняють умови вище.

Точності 10^{-6} можна досягти, використовуючи звичайну арифметику з рухомою комою подвійної точності (тип `double`).

L Три комп'ютери

Ліміт часу 3 с · Ліміт пам’яті 256 МіБ · Стандартний ввід/вивід

Команда з трьох студентів бере участь у незвичайній олімпіаді з програмування. Кожен учасник команди має власний комп'ютер, тож команда може працювати над трьома задачами одночасно.

В олімпіаді є n задач. Задачу i можна розв'язати, лише якщо один учасник працює над нею без перерви з хвилини l_i до хвилини r_i включно. Протягом однієї хвилини учасник може працювати лише над однією задачею, а ділити задачу між кількома учасниками не можна. Наприклад, один учасник може розв'язати задачі, яким потрібні хвилини 1–3 та 4–5, але не задачі, яким потрібні хвилини 1–3 та 3–5, бо обом потрібна хвилина 3.

За розв'язану задачу i команда отримує w_i балів. Знайдіть найбільшу сумарну кількість балів, яку може отримати команда.

Вхідні дані

Перший рядок містить одне ціле число t ($1 \le t \le 2 \cdot 10^5$) — кількість тестових випадків.

Перший рядок кожного тестового випадку містить одне ціле число n ($1 \le n \le 2 \cdot 10^5$) — кількість задач.

Кожен із наступних n рядків містить три цілі числа l_i, r_i і w_i ($1 \le l_i \le r_i \le 10^9$, $1 \le w_i \le 10^9$) — першу й останню хвилини задачі i та кількість балів за неї.

Гарантується, що сума n по всіх тестових випадках не перевищує $2 \cdot 10^5$.

Вихідні дані

Для кожного тестового випадку виведіть в окремому рядку одне ціле число — найбільшу сумарну кількість балів.

Приклади

Вхідні дані

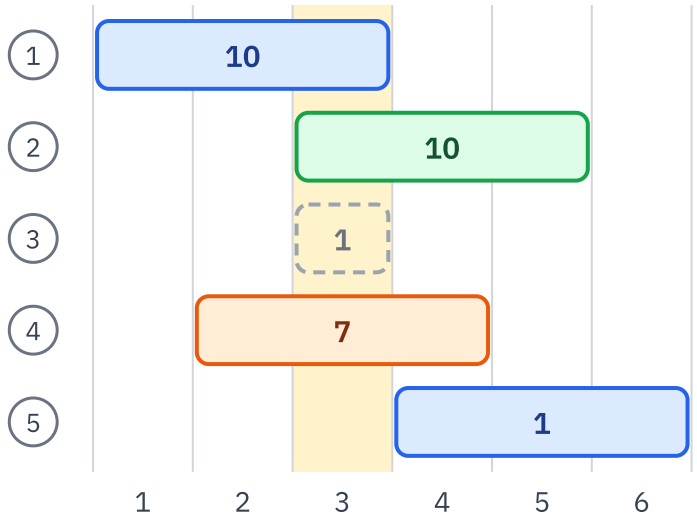
```
3
4
1 5 3
2 6 4
3 7 5
4 8 2
5
1 3 10
3 5 10
3 3 1
2 4 7
4 6 1
1
1000000000 1000000000 1000000000
```

Вихідні дані

```
12
28
1000000000
```

Примітка

У першому тестовому випадку всім чотирьом задачам потрібні хвилини 4 і 5, тож розв'язати можна щонайбільше три з них. Найвигідніше відмовитися від задачі, що приносить 2 бали, і отримати $3 + 4 + 5 = 12$ балів.



У другому тестовому випадку першим чотирьом задачам потрібна хвилина 3, тож від однієї з них доведеться відмовитися. Найвигідніше відмовитися від третьої задачі. Інші чотири задачі можна розв'язати: перший учасник розв'язує задачі 1 (хвилини 1–3) і 5 (хвилини 4–6), другий — задачу 2, а третій — задачу 4. Разом $10 + 1 + 10 + 7 = 28$ балів.

У третьому тестовому випадку єдину задачу можна розв'язати.

M Гірлянда UCPC

Ліміт часу 4,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Організатори UCPC прикрашають зал змагань довгою гірляндою з ламп у формі літер. Гірлянда складає рядок s довжини n з великих латинських літер: i-та лампа показує літеру s_i. Кожну лампу треба ввімкнути одним із трьох кольорів: синім, жовтим або білим.

Звісно ж, улюблене слово організаторів — `UCPC`. *Входженням* слова `UCPC` у гірлянду називається вибір чотирьох ламп на позиціях $i_1 < i_2 < i_3 < i_4$, які показують відповідно літери `U`, `C`, `P`, `C` (вибрані лампи не обов'язково стоять поруч). *Розфарбуванням входження* називається послідовність кольорів ламп i_1, i_2, i_3, i_4 саме в такому порядку.

Розфарбування всієї гірлянди називається *узгодженим*, якщо всі входження слова `UCPC` мають **однакове розфарбування**. Іншими словами, хоч як глядач прочитає слово `UCPC` у гірлянді, він завжди побачить ту саму послідовність кольорів. Зокрема, якщо `UCPC` узагалі не входить у гірлянду, то будь-яке розфарбування є узгодженим.

Порахуйте кількість узгоджених розфарбувань гірлянди. Два розфарбування вважаються різними, якщо хоча б одна лампа має в них різні кольори. Оскільки відповідь може бути великою, виведіть її за модулем 998 244 353.

Організатори розглядають кілька гірлянд, і відповісти на це запитання потрібно для кожної з них.

Вхідні дані

Перший рядок містить ціле число t ($1 \le t \le 10^6$) — кількість гірлянд.

Кожен із наступних t рядків містить рядок s ($1 \le |s| \le 10^6$, де $|s|$ — довжина рядка s) — гірлянду. Гарантується, що гірлянда складається з великих латинських літер.

Сумарна довжина всіх t рядків не перевищує 10^6.

Вихідні дані

Для кожної гірлянди виведіть в окремому рядку одне ціле число — кількість її узгоджених розфарбувань за модулем 998 244 353.

Приклади

Вхідні дані	Вихідні дані
3 UCPCPC UUCCPPCC CPU	27 81 27

Примітка

У першій гірлянді прикладу слово `UCPC` входить чотири рази, на позиціях $(1, 2, 3, 4)$, $(1, 2, 3, 6)$, $(1, 2, 5, 6)$ та $(1, 4, 5, 6)$. Лампа 4 є другою літерою останнього входження і четвертою літерою першого, тому обидві літери `C` слова мусять мати однаковий колір. Звідси всі лампи з літерою `C` мають один спільний колір, і обидві лампи з літерою `P` мають один спільний колір. Разом із кольором лампи 1 це дає $3^3 = 27$ узгоджених розфарбувань.

У другій гірлянді жодна лампа не може бути одночасно другою і четвертою літерою входження, тому дві літери `C` слова можуть мати різні кольори: лампи $1, 2$ мають один спільний колір,

лампи $3, 4$ — один спільний колір, лампи $5, 6$ — один спільний колір і лампи $7, 8$ — один спільний колір, що дає $3^4 = 81$ розфарбування.

У третій гірлянді слово `UCPC` не входить узагалі, тому всі $3^3 = 27$ розфарбувань узгоджені.

N Щоденник погоди

Ліміт часу 0,5 с · Ліміт пам'яті 256 МіБ · Стандартний ввід/вивід

Протягом n днів поспіль метеоролог-аматор Остап записував у щоденник, яким був кожен день: *сонячним* чи *дощовим*.

Молодший брат Остапа розлив на щоденник чай, і щоденні записи стали нечитабельними. Уцілiла лише сторінка з підсумком наприкінці. Для кожної пари сусідніх днів (день i і день $i+1$, де $1 \le i < n$) Остап порахував, якого вона типу:

- e пар, у яких за сонячним днем іде сонячний;
- g пар, у яких за дощовим днем іде дощовий;
- u пар, у яких за сонячним днем іде дощовий;
- v пар, у яких за дощовим днем іде сонячний.

Кожну з $n-1$ пар пораховано рівно один раз, тому $e+g+u+v=n-1$.

Остап не впевнений, що не помилився в підрахунках. Визначте, чи існує хоча б одна послідовність погоди на n днів, яка узгоджується з усіма чотирма числами підсумку.

У щоденнику є підсумки за кілька різних періодів, тож відповісти на це питання треба для t незалежних підсумків.

Вхідні дані

Перший рядок містить ціле число t ($1 \le t \le 10^5$) — кількість підсумків.

Кожен із наступних t рядків описує один підсумок і містить п'ять цілих чисел n, e, g, u, v ($1 \le n \le 2 \cdot 10^5$; $e, g, u, v \ge 0$; $e+g+u+v=n-1$).

Сума n за всіма підсумками не перевищує $4 \cdot 10^6$.

Вихідні дані

Для кожного підсумку виведіть окремий рядок: `YES`, якщо існує послідовність погоди, що йому відповідає, або `NO` в іншому разі.

Регістр літер не має значення: `yes`, `Yes` і `YES` приймаються однаково.

Приклади

Вхідні дані

```
4
5 1 1 1 1
5 1 0 2 1
5 0 0 4 0
3 0 0 1 1
```

Вихідні дані

```
YES
YES
NO
YES
```

Примітка

Позначимо сонячний день літерою `S`, а дощовий — літерою `R`.

У першому підсумку прикладу підходить послідовність `SSRRS`: її пари — `SS`, `SR`, `RR` і `RS`.

У другому підсумку підходить послідовність `SSRSR`: її пари — `SS`, `SR`, `RS` і `SR`.

У третьому підсумку всі чотири пари мали б бути «сонячний, потім дощовий». Тоді другий день мав би бути дощовим (як другий день першої пари) і водночас сонячним (як перший день

другої пари), а це неможливо.

У четвертому підсумку підходить послідовність SRS.