Загальна інформація

Загальне

- Вам дано 5 годин на рішення 5 задач, кожна з яких оцінюється в 100 балів.
- Ви можете зробити до 60 відправлень. Немає значення, як саме ви використаєте ці відправлення по задачах. Ви можете відправити будь-яку кількість рішень на кожну задачу, але сумарна кількість відправлень по всім задачам не має перевищувати 60.
- Журі не гарантує, що існують розв'язки на повний бал на таких мовах, як `Pascal`, `Python` та `Java`.
- Під час олімпіади суворо забороняється використовувати інтернет, за виключенням сайту, на якому ви працюєте. Не можна використовувати будь-які переносні носії інформації.
- Результати олімпіади будуть доступні на сайті `oi.in.ua` після змагання.

Оцінювання

Є два види оцінювання:

- «Потестове оцінювання». Кожний тест оцінюється незалежно від інших. Проходження тесту приносить певну кількість балів. Приклади оцінюються в 0 балів.
- «Блочне оцінювання». Усі тести поділені на блоки, які описані в умові задачі. Бали нараховуються лише при проходженні **всіх** тестів блоку. Якщо обмеження блока i не менші за обмеження блока j, то для нарахування балів за блок i, також потрібно, щоб пройшли всі тести блока j. В умові про це не буде сказано. Також є «нульовий блок», який складається з прикладів, він оцінюється в 0 балів. В умові про це згадувати не будуть.

Види задач

Є два види задач:

- «З вводом та виводом даних». Вам потрібно зчитати дані з файлу типу «`problem.in`», рішити задачу та вивести результат у файл «`problem.out`». Зверніть увагу, що ім'я файлів змінюється в залежності від задачі.
- «З модулями». Вам потрібно реалізувати функції, які описані у задачі. Зверніть увагу, що в цьому виді задач **суворо забороняється** зчитувати та виводити дані. Ви маєте працювати з даними лише в той спосіб, який описаний в умові задачі. Вам дадуть архів, в якому буде три файли для кожної мови програмування: header, sample, footer. Вам потрібно змінити sample та відправити лише його в систему. Під час компіляції вашого рішення, код в header вставляється перед вашим кодом, а код в footer після. Тобто формується новий файл, який починається з header, потім йде ваше рішення, і лише після нього footer. Зверніть увагу, що footer, який використовується при тестуванні, може відрізнятись від того, який буде в архіві.

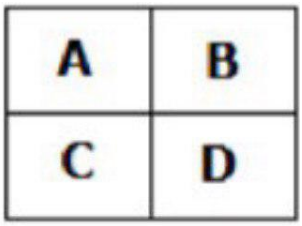
Питання

- Ви можете ставити питання за допомогою системи. Зверніть увагу, що питання мають бути такими, на які можна відповісти «Так» або «Ні». Ви можете отримати одну з наступних відповідей:
  - «Відповідь в умові», якщо на ваше питання можна відповісти, прочитавши умову або загальну інформацію про тур.
  - «Без коментарів», якщо питання стосується інформації, яку журі не бажає розголошувати, наприклад, методу розв'язання задачі.
  - «Незрозуміле питання», якщо журі не зрозуміло ваше питання.
  - «Так» або «Ні».

Задача A. Козак Вус і важлива знахідка

Назва вхідного файлу:	`lesson.in`
Назва вихідного файлу:	`lesson.out`
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

Зовсім нещодавно мешканці Потоколяндії знайшли стародавню табличку розміром 2×2, в якій розташовано чотири числа A, B, C і D так:

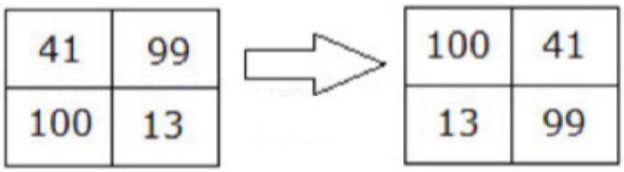


Вони відразу зрозуміли, що це дуже важлива історична знахідка. Спершу вони віднесли її Козаку Вусу для того, щоб він визначив важливість цієї таблички. На думку Козака Вуса, важливість таблички дорівнює $A \cdot (B + C - D)$.

На жаль, правильне положення таблички невідоме. Тому може статись таке, що однозначно визначити важливість таблички неможливо, оскільки це значення залежить від того, скільки разів її обернути.

Припустимо, що один оберт — це оберт за годинниковою стрілкою на 90°.

Наприклад, якщо $A = 41$, $B = 99$, $C = 100$, $D = 13$, то важливість рівна $41 \cdot (99 + 100 - 13) = 7\,626$. А якщо її один раз обернути, то $100 \cdot (41 + 13 - 99) = -4\,500$.



Козак Вус хоче з'ясувати, яку максимальну можливу важливість може мати ця знахідка. Але Вас він просить дізнатись, яку мінімальну кількість обертів потрібно зробити для того, щоб важливість таблички була максимальною.

Формат вхідних даних

Перший рядок містить чотири цілих числа A, B, C і D ($-10^8 \leqslant A, B, C, D \leqslant 10^8$) — числа на табличці.

Формат вихідних даних

Виведіть мінімальну кількість обертів, які потрібно зробити Козаку Вусу, щоб важливість таблички була максимальною.

Приклади

lesson.in	lesson.out
5 3 4 6	1
2 9 -4 13	3
2 6 3 0	0

Примітка

У першому прикладі спочатку табличка має важливість 5, але, якщо обернути її один раз, то вона набуде максимальної важливості, що рівна 32.

У другому прикладі достатньо трьох обертів, щоб табличка отримала своє максимальне значення важливості, яке рівне 171.

У останньому прикладі табличку повертати не треба, бо вона вже має максимальне значення важливості, яке рівне 18.

Оцінювання

Кожний тест, крім вхідних, оцінюється в 5 балів.

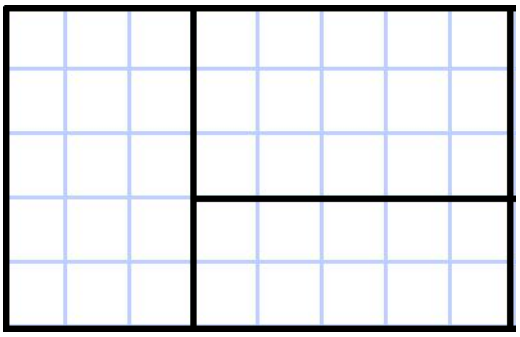
Задача B. Козак Вус і цікава задача

Назва вхідного файлу:	`room.in`
Назва вихідного файлу:	`room.out`
Обмеження використання часу:	0.25 seconds
Обмеження використання пам'яті:	256 megabytes

Всім відомо, що Козак Вус дуже захоплюється математикою. Сьогодні, читаючи книгу «Конкретна математика», він знайшов дуже цікаву задачу і вирішив запропонувати Вам її розв'язати.

Є кімната, яка має прямокутну форму. Одна з її сторін має довжину n, а друга — m. Вам необхідно з'ясувати, чи можливо розділити кімнату на **рівно три** окремі кімнати з цілими довжинами сторін так, щоб загальний периметр цих кімнат був рівно p.

Наприклад, якщо $n = 5$, $m = 8$ і $p = 46$, то один з можливих варіантів поділу кімнати такий:



Виведіть «YES» та розміри кімнат, якщо кімнату можливо розділити на три окремі кімнати, відповідно до умови задачі, інакше — «NO».

Формат вхідних даних

Перший рядок містить три цілих числа n, m та p ($1 \leqslant n, m \leqslant 10^9, 1 \leqslant p \leqslant 10^{15}$) — довжини сторін та периметр відповідно.

Формат вихідних даних

Виведіть «YES», якщо кімнату можливо розділити на три окремі кімнати з цілими довжинами сторін так, щоб загальний периметр цих кімнат був рівний p, інакше — «NO».

Якщо відповідь «YES», тоді у кожному з наступних трьох рядків треба вивести розміри відповідної кімнати. Розміри можна виводити у будь-якому порядку. Якщо відповідей декілька, то виведіть будь-яку.

Приклади

room.in	room.out
2 2 14	YES 2 1 1 1 1 1
2 3 17	NO
5 8 46	YES 5 3 2 5 3 5

Примітка

У першому прикладі можна поділити кімнату на три кімнати з розмірами 2×1, 1×1 та 1×1 відповідно.

У другому прикладі неможливо поділити кімнату на три із цілими сторонами, так щоб загальний периметр був рівний 17.

Третій приклад пояснено в умові.

Оцінювання

Кожний тест, крім прикладів, оцінюється від 1 до 4 балів.

Задача C. Козак Вус і НСД

Назва вхідного файлу:	`gcdarray.in`
Назва вихідного файлу:	`gcdarray.out`
Обмеження використання часу:	0.5 seconds
Обмеження використання пам'яті:	256 megabytes

Сьогодні Козак Вус зустрівся зі своїм давнім другом — Козаком Вухом. Вони дуже довго розмовляли, згадували своє дитинство та юність. Так і зайшла мова про задачу, яку колись вони не змогли вирішити на олімпіаді з програмування.

Дано масив з n чисел. За один хід можна одне з чисел **збільшити на** 1. Вам необхідно з'ясувати, за яку мінімальну кількість операцій можливо отримати масив, який буде задовольняти такі умови:

- $a_i \leqslant a_{i+1}$ для всіх i від 1 до $n-1$.
- найбільший спільний дільник усіх чисел більший за 1.

`Найбільший спільний дільник` множини додатних чисел — це найбільше додатне число, що одночасно є дільником усіх чисел з множини.

Формат вхідних даних

Перший рядок містить одне ціле число t ($1 \leqslant t \leqslant 5$) — кількість тестів. Далі слідує опис кожного тесту.

Перший рядок опису кожного тесту містить одне ціле число n ($1 \leqslant n \leqslant 10^4$) — розмір масиву.

Другий рядок опису кожного тесту містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^4$) — числа масиву.

Формат вихідних даних

Для кожного тесту в окремому рядку виведіть одне число — мінімальну кількість операцій, які необхідно виконати для того, щоб масив задовольняв дані умови.

Приклади

gcdarray.in	gcdarray.out
1 3 9 1 16	10
2 4 5 7 3 6 5 4 2 8 16 10	7 8

Примітка

У першому прикладі можна перше та друге число збільшити до 10, тоді найбільший спільний дільник чисел з масиву буде рівний два.

У першому тесті другого прикладу можна усі числа зробити рівними 7.

У другому тесті другого прикладу масив можна змінити до масиву $[4, 4, 8, 16, 16]$.

Оцінювання

№	Обмеження	Додаткові обмеження	Бали
1	$1 \leqslant n \leqslant 10^4$	Усі числа парні	10
2	$1 \leqslant n \leqslant 10$	-	20
3	$1 \leqslant n \leqslant 10^3$	-	30
4	$1 \leqslant n \leqslant 10^4$	-	40

Задача D. Козак Вус і Потоколяндія

Назва вхідного файлу:	`roads.in`
Назва вихідного файлу:	`roads.out`
Обмеження використання часу:	1 second
Обмеження використання пам'яті:	512 megabytes

У Потоколяндії є n будинків, у i-му з яких проживають a_i мешканців. Між цими будинками є m доріг, кожна дорога сполучає будинки v_i і u_i. Ми визначаємо `щастя` кожного мешканця як кількість мешканців (включно з собою), яких він може зустріти. Житель будинку може зустріти іншого жителя, якщо він з його будинку, або з будинку до якого можна дістатися, подорожуючи по дорогах Потоколяндії.

Останні d днів кожного дня відбувалася подія одного з двох типів:

1. Снігом засипало дорогу між будинками g_i і h_i, тож тепер по ній не можна проїхати.
2. k_i мешканців w_i-го будинку на вертольоті відправлялися у гості до далеких родичів за межами Потоколяндії.

Мешканці Потоколяндії пишуть листи Козаку Вусу з проханням повідомити їм останній такий день, що сума щастя будь-якого мешканця x_i-го будинку та будь-якого мешканця y_i-го будинку принаймні z_i.

Можна вважати, що всі дії відбуваються миттєво в першу мить кожного дня. Якщо і до початку всіх подій сумарне щастя менше за z_i, то потрібно вивести -1. Якщо сумарне щастя було не менше за z_i лише до першої події, то потрібно вивести 0. Якщо після i-ої події сумарне щастя стало меншим, ніж потрібно, то потрібно вивести $i - 1$. Якщо після всіх подій сумарне щастя принаймні z_i, то потрібно вивести n.

Оскільки Козак Вус досить зайнята людина, а мешканців Потоколяндії багато, він просить Вас допомогти йому відповісти на всі листи.

Формат вхідних даних

Перший рядок містить чотири цілих числа n, m, d та s ($1 \leqslant n, m, d, s \leqslant 2 \cdot 10^5$) — кількість будинків, доріг, днів та повідомлень відповідно.

Наступний рядок містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$) — кількість жителів в i-му будинку.

Кожний з наступних m рядків містить по два цілих числа v_i та u_i ($1 \leqslant v_i, u_i \leqslant n$, $u_i \neq v_i$) — будинки між якими є дорога. Гарантується, що немає кратних ребер.

Кожен з наступних d рядків описує запит в одному з двох форматів:

- «1 g_i h_i» ($1 \leqslant g_i, h_i \leqslant n$) — дорога, яку засипало снігом. Гарантується, що така дорога існує, і вона не була засипана снігом раніше.
- «2 w_i k_i» ($1 \leqslant w_i \leqslant n$, $1 \leqslant k_i \leqslant 10^9$) — номер будинку та кількість людей, які покинули його. Гарантується, що в будь-який момент часу в кожному будинку буде принаймні один мешканець.

Кожен з наступних s рядків містить по три цілих числа x_i, y_i та z_i ($1 \leqslant x_i, y_i \leqslant n$, $1 \leqslant z_i \leqslant 10^9$).

Формат вихідних даних

Виведіть s окремих рядків, у кожному з яких відповідь на відповідний запит. Якщо сума щастя двох мешканців ще до першого дня менша z, то виведіть -1.

Приклади

roads.in	roads.out
3 3 5 4 2 9 4 1 2 2 3 1 3 2 2 1 1 2 3 2 3 3 1 1 2 1 1 3 1 2 11 3 2 20 1 3 15 2 3 10	4 3 3 4
4 5 3 4 1 2 3 4 1 2 2 3 1 3 3 4 2 4 1 2 4 1 3 4 2 3 2 1 4 21 1 4 20 1 3 9 2 2 2	-1 1 2 3

Примітка

Пояснення другого прикладу:

Для першого запиту сумарне щастя двох мешканців завжди буде менше 21 (спочатку воно рівне 20). Для другого запиту, після другого дня їх сумарне щастя буде зменшено до $6 + 4 = 10$. Інші запити пояснюються так само.

Оцінювання

№	Обмеження				Бали
	n, m	d	s	Додаткові	
1	$1 \leqslant n, m \leqslant 200$	$1 \leqslant d \leqslant 200$	$1 \leqslant s \leqslant 200$	-	4
2	$1 \leqslant n, m \leqslant 2\,000$	$1 \leqslant d \leqslant 2\,000$	$1 \leqslant s \leqslant 2\,000$		7
3	$1 \leqslant n, m \leqslant 2 \cdot 10^5$				13
4	$1 \leqslant n, m \leqslant 5\,000$	$1 \leqslant d \leqslant 5\,000$	$1 \leqslant s \leqslant 2 \cdot 10^5$		14
5	$1 \leqslant n, m \leqslant 2 \cdot 10^5$	$1 \leqslant d \leqslant 2 \cdot 10^5$		$x_i = y_i$	27
6				-	35

Задача E. Козак Вус і свята

Обмеження використання часу:	2 seconds
Обмеження використання пам'яті:	512 megabytes

Як відомо, мешканці царства Потоколяндія — дуже педантичні люди. І навіть коли справа доходить до свят, вони завжди хочуть бути впевненими в тому, що все пройде дуже добре. Тому розклад всіх свят складений на сто років вперед. Козак Вус вирішив запросити свого друга — Козака Вуха приїхати в одне із міст царства і відвідати як можна більше свят.

У царстві n міст, які сполучені $n-1$ двонаправленою дорогою так, що з будь-якого міста можна дістатися до іншого, можливо, відвідуючи інші міста. Для того, щоб пройти по i-й дорозі, потрібно l_i днів.

Кожне свято в Потоколяндії характеризується номером міста c_i, в якому воно буде проходити, і номером дня d_i, в який буде відбуватися свято. Козак Вух не витрачає багато часу на святкування. Тому, якщо він святкує в i-ий день, то він може в той самий день виїхати, приїхати в інше місто наступного дня (якщо є така дорога, що $l_i = 1$) та святкувати (якщо таке свято є).

Друг Козака Вуса — такий щасливчик, що день його прибуття до царства має номер 0 в календарі, причому спочатку він може приїхати в будь-яке місто царства. Козак Вус вирішив дізнатися, яку максимальну кількість свят може відвідати його друг. Для цього він звернувся за допомогою до Вас. Допоможіть йому це зробити!

Протокол взаємодії

Вам потрібно реалізувати функцію (тут використовується **псевдокод**, щоб дізнатись деталі реалізації для вашої мови, дивіться нижче):

```
integer solve(integer n, integer m, array of integers e1, array of integers e2,
              array of integers len, array of integers q1, array of integers q2)
```

- n — кількість міст у царстві;
- m — кількість свят у царстві;
- $e1$ — масив довжини $n-1$, що задає перший кінець відповідного ребра;
- $e2$ — масив довжини $n-1$, що задає другий кінець відповідного ребра;
- len — масив довжини $n-1$, що задає довжину відповідного ребра;
- $q1$ — масив довжини m, що задає номер міста, у якому буде проходити відповідне свято;
- $q2$ — масив довжини m, що задає день, у який буде проходити відповідне свято.

Зверніть увагу, що індексація в масивах починається з нуля, не з одиниці.

В залежності від того, на якому сервері ви пишете, ви можете завантажити архів за одним з наступних посилань:

- http://kremped.org.ua/files/uoi-2019-region-tour-1-9qeroafk2.zip
- http://ejudge.sumdu.edu.ua/statements/uoi-2019-region-tour-1-oek19spa.zip
- http://olymp.uzhnu.edu.ua/statements/uoi-2019-region-tour-1-9qks0gma.zip
- http://dn.hoippo.km.ua/statements/tur1/122f4058466cff9a5b50c5a3628259e3.zip

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leqslant n \leqslant 2 \cdot 10^5$) — кількість міст у царстві.

Кожен з наступних $n-1$ рядків містить по три цілих числа a_i, b_i та l_i ($1 \leqslant a_i, b_i \leqslant n$, $1 \leqslant l_i \leqslant 10^9$) — номера міст, які з'єднує дорога, і кількість днів, яка необхідна для її подолання. Гарантується, що граф зв'язний.

Наступний рядок містить одне ціле число m ($1 \leqslant m \leqslant 2 \cdot 10^5$) — кількість свят у царстві.

Кожен з наступних m рядків містить по два цілих числа c_i і d_i ($1 \leqslant c_i \leqslant n$, $1 \leqslant d_i \leqslant 10^9$) — номер міста та номер дня, в який буде відбуватися i-те свято.

Формат вихідних даних

Виведіть одне число — максимальну кількість свят, які зможе відвідати Козак Вух.

Приклади

input	output
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3
11 2 1 2 3 2 5 4 1 5 5 2 4 6 5 1 7 1 2 8 3 4 9 6 2 10 7 2 11 2 2 9 1 67 1 34 11 16 5 97 4 70 2 20 2 61 2 26 2 70	8

Примітка

Спочатку Козак Вух може прибути до міста 3 та зачекати один день до святкування. Після цього, в перший день він може за два дні переїхати до міста 1, де у третій день буде свято. Так само у третій день він може виїхати до міста 2, де у четвертий день також буде свято. Але до останнього свята він вже не встигне доїхати, бо потрібно 3 дні, щоб дібратись до міста 4. Таким чином, він відвідав 3 свята.

Оцінювання

№	Обмеження			Бали
	n	m	l_i, d_i	
1	$1 \leqslant n \leqslant 100$	$1 \leqslant m \leqslant 9$	$1 \leqslant l_i, d_i \leqslant 100$	14
2	$1 \leqslant n \leqslant 2\,000$	$1 \leqslant m \leqslant 2\,000$	$1 \leqslant l_i, d_i \leqslant 5\,000$	17
3	$1 \leqslant n \leqslant 5\,000$	$1 \leqslant m \leqslant 5\,000$	$1 \leqslant l_i, d_i \leqslant 10^9$	28
4	$1 \leqslant n \leqslant 10^5$	$1 \leqslant m \leqslant 10^5$		22
5	$1 \leqslant n \leqslant 2 \cdot 10^5$	$1 \leqslant m \leqslant 2 \cdot 10^5$		19

Деталі реалізації для C

Реалізуйте одну функцію:

```
int solve(int n, int m, int e1[], int e2[], int l[], int c[], int d[])
```

Деталі реалізації для C++

Реалізуйте одну функцію:

```
int solve(int n, int m, vector<int> e1, vector<int> e2, vector<int> len,
          vector<int> q1, vector<int> q2)
```

Деталі реалізації для Java

Вам потрібно реалізувати `class Test`, в якому має бути одна функція:

```
public static int solve(int n, int m, List<Integer> v, List<Integer> u, List<Integer> len,
                        List<Integer> c, List<Integer> d)
```

Деталі реалізації для Python

Реалізуйте одну функцію:

```
def solve(n, m, v, u, l, c, d):
```

Деталі реалізації для Pascal

Реалізуйте одну функцію:

```
Function solve(n, m: Integer; e1, e2, l, c, d: array of Integer):Integer;
```