

## С. Про покраску забора

ограничение по времени на тест: 1 секунда  
 ограничение по памяти на тест: 512 мегабайт

Бизон-Чемпион не только внимательный, но и очень трудолюбивый.

Бизон-Чемпион решил покрасить свой старый забор в свой любимый оранжевый цвет. Забор представляет из себя  $n$  вертикальных досок, расположенных в ряд. Между соседними досками нет зазора. Доски нумеруются слева направо начиная с единицы,  $i$ -я доска имеет ширину 1 метр и высоту  $a_i$  метров.

Бизон-Чемпион купил в магазине кисть шириной 1 метр. Он умеет делать вертикальные и горизонтальные мазки этой кистью. Причем во время мазка кисть должна непрерывно и полностью касаться забора (смотрите пояснения примеров для лучшего понимания). Какое минимальное количество мазков придется сделать Бизону-Чемпиону, чтобы полностью покрасить забор? Обратите внимание, что разрешается красить один и тот же участок забора несколько раз.

### Входные данные

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 5000$ ) — количество досок забора. Во второй строке записаны через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Выходные данные

Выведите целое число — минимальное количество мазков, необходимых для покраски всего забора.

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5<br>2 2 1 2 1         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 3                      |                             |
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 2<br>2 2               |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 2                      |                             |
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 1<br>5                 |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 1                      |                             |

### Примечание

В первом примере можно покрасить забор за три мазка кистью: первый проходит на высоте 1 горизонтально по всем доскам, второй — на высоте 2 горизонтально красит первую и вторую доски, а третий мазок (он может быть как горизонтальный, так и вертикальный) докрашивает четвертую доску.

Во втором примере можно покрасить забор как двумя горизонтальными, так и двумя вертикальными мазками.

В третьем примере всего одна доска, которую можно покрасить, используя один вертикальный мазок.

## D. Феникс и наука

ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт

Феникс решил стать ученым! Сейчас он изучает рост бактерий.

Изначально, в день 1, есть одна бактерия массой 1.

Каждый день некоторое количество бактерий делятся (возможно, все или ни одной). Когда бактерия массой  $m$  делится, она превращается в две бактерии массой  $\frac{m}{2}$  каждая. Например, бактерия массой 3 может разделиться на две бактерии массой 1.5.

Каждую ночь масса каждой бактерии увеличивается на один.

Феникса интересует вопрос: может ли общая масса бактерий стать в точности равна  $n$ . Если может, то его интересует как добиться такой массы за минимальное количество ночей. Помогите ему стать самым лучшим ученым!

### Входные данные

Входные данные состоят из нескольких наборов. В первой строке задано целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных.

В единственной строке каждого набора задано целое число  $n$  ( $2 \leq n \leq 10^9$ ) — суммарная масса бактерий, которой хочет добиться Феникс.

### Выходные данные

Для каждого набора входных данных, если не существует способа получить общую массу равную  $n$ , выведите -1. В противном случае, выведите две строки.

В первой строке выведите целое число  $d$  — минимальное необходимое количество ночей.

В следующей строке выведите  $d$  целых чисел, где  $i$ -е число обозначает количество бактерий, которые поделятся в  $i$ -й день.

Если существует несколько решений, выведите любое из них.

### Пример

| входные данные                     | Скопировать |
|------------------------------------|-------------|
| 3<br>9<br>11<br>2                  |             |
| выходные данные                    | Скопировать |
| 3<br>1 0 2<br>3<br>1 1 2<br>1<br>0 |             |

### Примечание

В первом наборе, можно получить общую массу бактерий ровно 9 следующим образом:

- День 1: Бактерия с массой 1 делится. Получаем две бактерии с массой 0.5.
- Ночь 1: Все бактерии увеличивают массу на один. Получаем две бактерии с массой 1.5.
- День 2: Ни одна бактерия не делится.
- Ночь 2: Получаем две бактерии с массой 2.5.
- День 3: Обе бактерии делятся. Получаем четыре бактерии с массой 1.25.
- Ночь 3: Получаем четыре бактерии с массой 2.25.

Суммарная масса равна  $2.25 + 2.25 + 2.25 + 2.25 = 9$ . Можно доказать, что 3 — минимальное необходимое количество ночей. Есть и другие способы добиться суммарной массы 9 за 3 ночи.

В втором наборе, можно получить общую массу бактерий ровно 11 следующим образом:

- День 1: Бактерия с массой 1 делится. Получаем две бактерии с массой 0.5.

- Ночь 1: Получаем две бактерии с массой 1.5.
- День 2: Одна бактерия делится. Получаем три бактерии с массами 0.75, 0.75 и 1.5.
- Ночь 2: Получаем три бактерии с массами 1.75, 1.75 и 2.5.
- День 3: Бактерия с массой 1.75 и бактерия с массой 2.5 делятся. Получаем пять бактерий с массами 0.875, 0.875, 1.25, 1.25 и 1.75.
- Ночь 3: Получаем пять бактерий с массами 1.875, 1.875, 2.25, 2.25 и 2.75.

Суммарная масса равна  $1.875 + 1.875 + 2.25 + 2.25 + 2.75 = 11$ . Можно доказать, что 3 — минимальное необходимое количество ночей. Есть и другие способы добиться суммарной массы 11 за 3 ночи.

Во третьем наборе, бактерия не делится в день 1, а потом увеличивает массу до 2 в ночь 1.

---

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:39:12<sup>UTC+5</sup> (i1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



**ІТМО**

## C. Gargari и слоны

ограничение по времени на тест: 3 секунды  
ограничение по памяти на тест: 256 мегабайт

Gargari завидует Caissa, потому что последний прошел игру из прошлой задачи. Теперь Gargari хочет доказать всем, что он самый умный.

У Gargari есть шахматная доска размером  $n \times n$ . В каждой клетке шахматной доски записано целое число. Gargari хочет разместить на шахматной доске два слона таким образом, что не существует клетки, которая находится под ударом обоих слонов. Рассмотрим клетку доски с числом  $x$ , записанным на ней, если она находится под ударом одного из слонов, тогда Gargari получает  $x$  долларов за эту клетку. Подскажите Gargari, как разместить двух слонов на доске, чтобы получить как можно больше денег.

Считается, что клетка находится под ударом слона, если она находится на одной диагонали с ним (клетка, в которой стоит слон, также считается находящейся под ударом).

### Входные данные

В первой строке записано целое число  $n$  ( $2 \leq n \leq 2000$ ). В каждой из следующих  $n$  строк записано  $n$  целых чисел  $a_{ij}$  ( $0 \leq a_{ij} \leq 10^9$ ) — описание шахматной доски.

### Выходные данные

В первой строке выведите максимальное количество долларов, которое Gargari может заработать. Затем выведите четыре целых числа:  $x_1, y_1, x_2, y_2$  ( $1 \leq x_1, y_1, x_2, y_2 \leq n$ ), где  $x_i$  обозначает номер строки, в которой находится  $i$ -й слон,  $y_i$  обозначает номер столбца, в котором находится  $i$ -й слон. Считайте, что строки шахматной доски пронумерованы от 1 до  $n$  сверху вниз, в столбцы пронумерованы от 1 до  $n$  слева направо.

Если есть несколько оптимальных решений, разрешается вывести любое.

### Примеры

| входные данные                                | Скопировать |
|---|-------------|
| 4<br>1 1 1 1<br>2 1 1 0<br>1 1 1 0<br>1 0 0 1 |             |
| выходные данные                               | Скопировать |
| 12<br>2 2 3 2                                 |             |



## D. СоздатеЛи

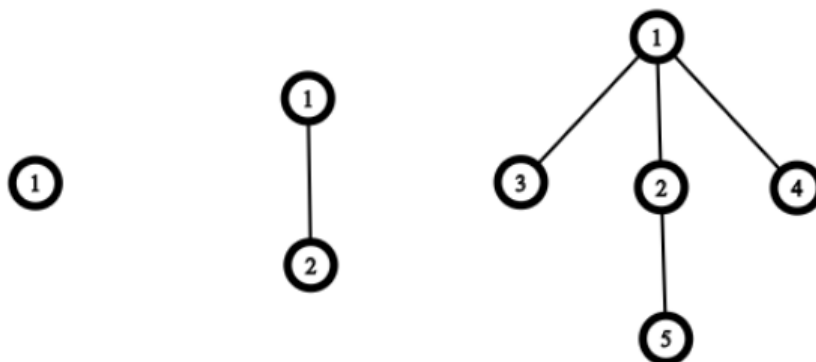
ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт

Ли потратил так много времени на создание хорошей  $\text{div.2 } D$  задачи, чтобы сбалансировать недавний контекст, но задача продолжает ощущаться неподходящей. Ли придумывал ее так мучительно долго, что заработал фобию  $\text{div.2 } D$  задач. И теперь он прячется в кустах...

Назовем *Корневым Сухим Кустом* (КСК) уровня  $n$  корневое дерево, построенное согласно правилам ниже.

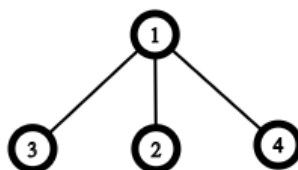
Корневой Сухой Куст уровня 1 — это одна вершина. Для построения КСК уровня  $i$ , сначала построим КСК уровня  $i - 1$  и далее для каждой вершины  $u$ :

- если у  $u$  нет детей, то подвесим к ней одного сына;
- если у  $u$  есть один сын, то подвесим к ней еще двух детей;
- если у  $u$  есть более одного сына, то пропустим ее.



Корневые Сухие Кусты уровня 1, 2 и 3.

Назовем *лапой* корневое дерево из четырех вершин: одна корневая вершина (также называется центром) и три ребенка. Оно напоминает лапу:



Центром лапы является вершина с номером 1.

У Ли есть Корневой Сухой Куст уровня  $n$ . Первоначально все вершины КСК зеленого цвета.

За один шаг, он может выбрать лапу в КСК и, если все вершины в ней *зеленые* и все вершины лапы являются детьми ее центра, покрасить вершины лапы в желтый.

Ли хочет узнать максимальное количество желтых вершин, которое он сможет получить. Так как ответ может быть очень большим, выведите его по модулю  $10^9 + 7$ .

**Входные данные**

В первой строке задано одно число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных.

В следующих  $t$  строках заданы сами наборы — по одному в строке.

В первой строке каждого набора задано одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^6$ ) — уровень КСК Ли.

**Выходные данные**

Для каждого набора входных данных выведите единственное целое число — максимальное количество желтых вершин, которые может получить Ли, по модулю  $10^9 + 7$ .

Пример

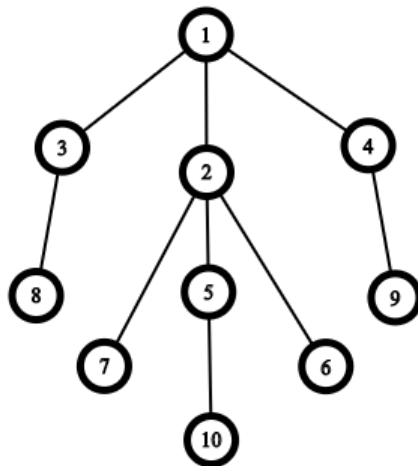
| входные данные                                   | Скопировать |
|--|-------------|
| 7<br>1<br>2<br>3<br>4<br>5<br>100<br>2000000     |             |
| выходные данные                                  | Скопировать |
| 0<br>0<br>4<br>4<br>12<br>990998587<br>804665184 |             |

Примечание

Несложно заметить, что ответ для КСК уровня 1 или 2 равен 0.

Ответ для КСК уровня 3 равен 4, так как есть только одна лапа, которую можно выбрать: {1, 2, 3, 4}.

Ответ для КСК уровня 4 равен 4, так как мы можем выбрать либо одну лапу {1, 3, 2, 4} или одну лапу {2, 7, 5, 6}. Других лап в КСК уровня 4 нет (например, мы не можем выбрать {2, 1, 7, 6}, так как 1 не является ребенком вершины-центра 2).



Корневой Сухой Куст уровня 4.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:39:14<sup>UTC+5</sup> (11).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

## D. Лучший отпуск

ограничение по времени на тест: 2 секунды  
 ограничение по памяти на тест: 256 мегабайт

Вы давно влюблены в Коронавирус-тян, но до сих пор не знали, где она живёт. И буквально сейчас выяснили, что она живёт в далёком местечке под названием Няху.

Вы немедленно решили взять отпуск на своей скучной работы и навестить Коронавирус-тян. Ваш отпуск длится ровно  $x$  дней, и ровно столько дней вы проведёте в гостях у подруги. В гостях вы проведёте ровно  $x$  подряд идущих дней.

В Няху используется очень необычный календарь: всего в году  $n$  месяцев,  $i$ -й месяц длится ровно  $d_i$  дней. Дни в  $i$ -м месяце пронумерованы от 1 до  $d_i$ . В Няху нет високосных годов.

Настроение Коронавирус-тян (и, соответственно, её желание вас обнять) зависит от номера дня в месяце. А именно, в  $j$ -й день месяца вы получаете  $j$  объятий, если вы находитесь в гостях у Коронавирус-тян.

Вы знаете об этой особенности своей подруги и хотите спланировать свою поездку так, чтобы суммарно получить наибольшее возможное количество объятий (и тогда, быть может, вы сможете завоевать сердечко Коронавирус-тян).

Обратите внимание, ваша поездка **не обязательно** должна начинаться и закончиться в одном и том же году.

### Входные данные

В первой строке входных данных даны два натуральных числа  $n$  и  $x$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество месяцев в году и количество дней, которые вы можете провести в гостях у подруги.

Во второй строке даны  $n$  целых чисел  $d_1, d_2, \dots, d_n$ , где  $d_i$  — количество дней в  $i$ -м месяце ( $1 \leq d_i \leq 10^6$ ).

Гарантируется, что  $1 \leq x \leq d_1 + d_2 + \dots + d_n$ .

### Выходные данные

Выведите единственное число — максимальное число объятий, которое вы можете получить от Коронавирус-тян за время лучшего отпуска в своей жизни.

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 3 2<br>1 3 1           |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 5                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 3 6<br>3 3 3           |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 12                     |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5 6<br>4 2 3 1 3       |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 15                     |                             |

### Примечание

В первом примере номера дней года (номера в соответствующих месяцах) имеет вид:  $\{1, 1, 2, 3, 1\}$ . Коронавирус-тян обнимет вас больше всего раз, если вы приедете в третий день года:  $2 + 3 = 5$  объятий.

Во втором примере номера дней года (номера в соответствующих месяцах) имеет вид:  $\{1, 2, 3, 1, 2, 3, 1, 2, 3\}$ . Больше всего объятий вы получите если приедете в третий день года:  $3 + 1 + 2 + 3 + 1 + 2 = 12$  объятий.

В третьем примере номера дней имеют вид:  $\{1, 2, 3, 4, 1, 2, 1, 2, 3, 1, 1, 2, 3\}$ . Больше всего объятий вы получите если приедете в двенадцатый день года: подруга обнимет вас  $2 + 3 + 1 + 2 + 3 + 4 = 15$  раз.

## D. Шичикуджи и электросеть

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Шичикуджи — новое местное божество Южного Храма Черной Улитки. Ее первое задание заключается в следующем:

В префектуре X построено  $n$  новых городов. Города пронумерованы от 1 до  $n$ . Город  $i$  находится в  $x_i$  км на север от храма и в  $y_i$  км на восток от храма. Возможно, что  $(x_i, y_i) = (x_j, y_j)$  даже если  $i \neq j$ .

Шичикуджи планирует обеспечить электричеством каждый город либо с помощью установки там электростанции, либо с помощью соединения его с другим городом, в котором уже есть электричество. То есть в городе есть электричество, если в нём есть электростанция или он соединён напрямую или по цепочке кабелей с городом, в котором есть электростанция.

- Возведение электростанции в городе  $i$  стоит  $c_i$  иен;
- Соединение города  $i$  и города  $j$  стоит  $k_i + k_j$  иен за **каждый км кабеля**, использованного для соединения. Однако кабель можно прокладывать вдоль сторон света (на север, юг, восток, запад). Кабели могут пересекать друг друга. У каждого кабеля оба конца должны быть в некоторых городах. Если город  $i$  и город  $j$  соединены кабелем, то кабель всегда будет прокладываться по кратчайшему пути из города  $i$  в город  $j$ . Тогда длина кабеля, соединяющего город  $i$  и город  $j$ , равна  $|x_i - x_j| + |y_i - y_j|$  км.

Шичикуджи хочет осуществить эту затею, потратив как можно меньше денег, так как она считает, что нет ничего важнее в мире, чем деньги. Однако она умерла, когда была всего в пятом классе (разумеется, божества не стареют после смерти), поэтому недостаточно смысленная, чтобы воплотить проект в жизнь. Поэтому она просит вас о помощи.

Таким образом, от вас требуется предоставить Шичикуджи следующую информацию: минимальное количество иен, необходимое, чтобы обеспечить электричеством все города, города, в которых будут построены электростанции, и проведенные соединения.

Если существует несколько способов так выбрать города и соединения, чтобы получить конструкцию минимальной цены, то выведите любую из них.

### Входные данные

В первой строке записано одно целое число  $n$  ( $1 \leq n \leq 2000$ ) — количество городов.

Затем следует  $n$  строк. В  $i$ -й строке записаны два целых числа  $x_i$  ( $1 \leq x_i \leq 10^6$ ) и  $y_i$  ( $1 \leq y_i \leq 10^6$ ) — координаты  $i$ -го города.

В следующей строке записаны  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 10^9$ ) — цена установки электростанции в  $i$ -м городе.

В последней строке записаны  $n$  целых чисел  $k_1, k_2, \dots, k_n$  ( $1 \leq k_i \leq 10^9$ ).

### Выходные данные

В первой строке выведите одно целое число, обозначающее минимальное количество иен, требуемых для конструкции.

Затем выведите одно целое число  $v$  — количество электростанций, которые требуется построить.

Далее выведите  $v$  целых чисел, обозначающих номера городов, в которых требуется построить электростанции. Каждое число должно быть от 1 до  $n$  и все числа должны быть попарно различны. Можно выводить числа в произвольном порядке.

После этого выведите одно целое число  $e$  — количество необходимых соединений.

Наконец, выведите  $e$  пар целых чисел  $a$  и  $b$  ( $1 \leq a, b \leq n, a \neq b$ ), обозначающих, что между городами  $a$  и  $b$  будет проведено соединение. Каждая неупорядоченная пара городов может встречаться не более одного раза (для каждого  $(a, b)$  не должно больше существовать пар  $(a, b)$  или  $(b, a)$ ). Пары можно выводить в произвольном порядке.

Если существует несколько способов так выбрать города и соединения, чтобы получить конструкцию минимальной цены, то выведите любую из них.

### Примеры

| входные данные                           | Скопировать |
|--|-------------|
| 3<br>2 3<br>1 1<br>3 2<br>3 2 3<br>3 2 3 |             |



выходные данные

Скопировать

8  
3  
1 2 3  
0

входные данные

Скопировать

3  
2 1  
1 2  
3 3  
23 2 23  
3 2 3

выходные данные

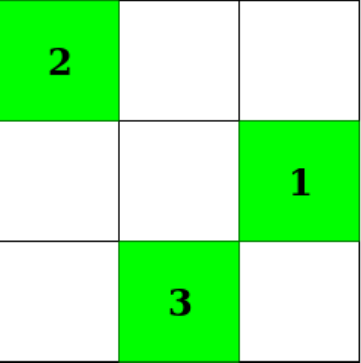
Скопировать

27  
1  
2  
2  
1 2  
2 3

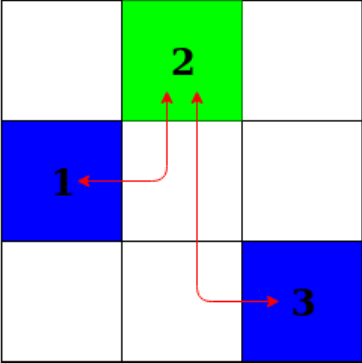
Примечание

Для ответов на примеры смотрите на данные графики (города с электростанциями раскрашены зеленым, остальные — синим, кабели покрашены в красный):

Example 1



Example 2



В первом примере цена строительства электростанций во всех городах равна  $3 + 2 + 3 = 8$ . Можно показать, что больше никакая конфигурация не стоит меньше 8 иен.

Во втором примере цена строительства электростанции в городе 2 равна 2. Стоимость соединения городов 1 и 2 равна  $2 \cdot (3 + 2) = 10$ . Стоимость соединения городов 2 и 3 равна  $3 \cdot (2 + 3) = 15$ . Поэтому итоговая цена равна  $2 + 10 + 15 = 27$ . Можно показать, что никакая конфигурация не стоит меньше 27 иен.



## D. Dr. Evil Underscores

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Сегодня Багри в качестве дружеского подарка вручил Бадави  $n$  целых чисел  $a_1, a_2, \dots, a_n$  и дал ему задачу выбрать такой  $X$ , что  $\max_{1 \leq i \leq n} (a_i \oplus X)$  принимает минимальное возможное значение, где  $\oplus$  обозначает операцию [побитового исключающего ИЛИ](#).

Бадави, как обычно, ленится, поэтому вы решили помочь ему и найти минимальное возможное значение  $\max_{1 \leq i \leq n} (a_i \oplus X)$ .

### Входные данные

В первой строке записано одно целое число  $n$  ( $1 \leq n \leq 10^5$ ).

Во второй строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 2^{30} - 1$ ).

### Выходные данные

Выведите одно целое число — минимальное возможное значение  $\max_{1 \leq i \leq n} (a_i \oplus X)$ .

### Примеры

|                        |             |
|------------------------|-------------|
| <b>входные данные</b>  | Скопировать |
| 3<br>1 2 3             |             |
| <b>выходные данные</b> | Скопировать |
| 2                      |             |

|                        |             |
|------------------------|-------------|
| <b>входные данные</b>  | Скопировать |
| 2<br>1 5               |             |
| <b>выходные данные</b> | Скопировать |
| 4                      |             |

### Примечание

В первом примере можно выбрать  $X = 3$ .

Во втором примере можно выбрать  $X = 5$ .



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

## G. A/B матрица

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам даны четыре целых положительных числа  $n, m, a, b$  ( $1 \leq b \leq n \leq 50$ ;  $1 \leq a \leq m \leq 50$ ). Найдите любую такую прямоугольную матрицу размера  $n \times m$ , что выполняются все следующие условия:

- в каждой строке матрицы ровно  $a$  единиц;
- в каждом столбце матрицы ровно  $b$  единиц;
- все остальные элементы — нули.

Если искомой матрицы не существует, то укажите это.

Например, для  $n = 3, m = 6, a = 2, b = 1$  существует такая матрица, удовлетворяющая условиям выше:

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

### Входные данные

В первой строке записано целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов тестовых данных в тесте. Далее следуют  $t$  наборов тестовых данных.

Каждый набор тестовых данных описывается четырьмя целыми положительными числами  $n, m, a, b$  ( $1 \leq b \leq n \leq 50$ ;  $1 \leq a \leq m \leq 50$ ), где  $n$  и  $m$  — размеры матрицы, а  $a$  и  $b$  — ограничения на количества единиц для строк и столбцов, соответственно.

### Выходные данные

Для каждого набора тестовых данных выведите:

- «YES» (без кавычек) и искомую матрицу (если существует несколько ответов, выведите любой), если она существует, или,
- «NO» (без кавычек), если ее не существует.

Чтобы вывести матрицу  $n \times m$ , выведите  $n$  строк, каждая из которых содержит  $m$  чисел 0 или 1, описывающих очередную строку матрицы. Числа нужно выводить **без пробелов**.

### Пример

| входные данные   | Скопировать |
|--|-------------|
| 5<br>3 6 2 1<br>2 2 2 1<br>2 2 2 2<br>4 4 2 2<br>2 1 1 2   |             |
| выходные данные  | Скопировать |
| YES<br>010001<br>100100<br>001010<br>NO<br>YES<br>11<br>11<br>YES<br>1100<br>1100<br>0011<br>0011<br>YES<br>1<br>1 |             |

## Е. К-периодичная гирлянда

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт

Вам задана гирлянда, состоящая из  $n$  ламп. Состояния ламп описываются строкой  $s$  длины  $n$ .  $i$ -й символ строки  $s_i$  равен '0', если  $i$ -я лампа выключена, или '1', если  $i$ -я лампа включена. Вам также задано положительное целое число  $k$ .

За один ход вы можете выбрать **одну лампу** и изменить ее состояние (то есть включить ее, если она выключена, и наоборот).

Гирлянда называется  $k$ -периодичной, если расстояние между **каждой парой соседних включенных ламп** равно **в точности**  $k$ . Рассмотрим случай  $k = 3$ . Тогда гирлянды «00010010», «1001001», «00010» и «0» являются хорошими, а гирлянды «00101001», «1000001» и «01001100» не являются хорошими. Заметьте, что **гирлянда не является цикличной**, то есть первая включенная лампа не идет после последней включенной лампы и наоборот.

Ваша задача — найти **минимальное** количество ходов, необходимое для того, чтобы получить  $k$ -периодичную гирлянду из заданной.

Вам необходимо ответить на  $t$  независимых наборов тестовых данных.

### Входные данные

Первая строка входных данных содержит одно целое число  $t$  ( $1 \leq t \leq 25\,000$ ) — количество наборов тестовых данных. Затем следуют  $t$  наборов тестовых данных.

Первая строка набора тестовых данных содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 10^6$ ;  $1 \leq k \leq n$ ) — длину  $s$  и необходимый период. Вторая строка набора тестовых данных содержит строку  $s$ , состоящую из  $n$  символов '0' и '1'.

Гарантируется, что сумма  $n$  по всем наборам тестовых данных не превосходит  $10^6$  ( $\sum n \leq 10^6$ ).

### Выходные данные

Для каждого набора тестовых данных выведите ответ — **минимальное** количество ходов, необходимое для того, чтобы получить  $k$ -периодичную гирлянду из заданной.

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 6<br>9 2<br>010001010<br>9 3<br>111100000<br>7 4<br>1111111<br>10 3<br>1001110101<br>1 1<br>1<br>1 1<br>0 |             |
| выходные данные   | Скопировать |
| 1<br>2<br>5<br>4<br>0<br>0  |             |

## С. Никита и ТЧ

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Никита — студент, увлеченный теорией чисел и алгоритмами. Он столкнулся с интересной задачей, связанной с массивом чисел.

Допустим, у Никиты есть массив целых чисел  $a$  длины  $n$ . Назовём подпоследовательность<sup>†</sup> массива *особенной*, если её **наименьшее общее кратное (НОК)** не содержится в  $a$ . НОК пустой подпоследовательности равен 0.

Никита задался вопросом: какова длина самой длинной *особенной* подпоследовательности массива  $a$ ? Помогите ему ответить на этот важный вопрос!

<sup>†</sup> Последовательность  $b$  является подпоследовательностью  $a$ , если  $b$  может быть получена из  $a$  путем удаления нескольких (возможно, нуля или всех) элементов, не изменяя порядок оставшихся элементов. Например,  $[5, 2, 3]$  является подпоследовательностью  $[1, 5, 7, 8, 2, 4, 3]$ .

### Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 2000$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 2000$ ) — длину массива  $a$ .

Вторая строка каждого набора входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива  $a$ .

Гарантируется, что сумма  $n$  по всем наборам входных данных не превышает 2000.

### Выходные данные

Для каждого набора выведите одно целое число — максимальную длину *особенной* подпоследовательности  $a$ .

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 6<br>5<br>1 2 4 8 16<br>6<br>3 2 10 20 60 1<br>7<br>2 3 4 6 12 100003 1200036<br>9<br>2 42 7 3 6 7 7 1 6<br>8<br>4 99 57 179 10203 2 11 40812<br>1<br>1 |             |
| выходные данные   | Скопировать |
| 0<br>4<br>4<br>5<br>8<br>0  |             |

### Примечание

В первом наборе входных данных НОК любой непустой подпоследовательности будет содержаться в  $a$ , поэтому ответ 0.

Во втором наборе входных данных можно взять подпоследовательность  $[3, 2, 10, 1]$ , её НОК — число 30, которое не содержится в  $a$ .

В третьем наборе входных данных можно взять подпоследовательность  $[2, 3, 6, 100\,003]$ , её НОК — число 600 018, которое не содержится в  $a$ .

## D. Заполни сумку

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

У вас есть сумка размера  $n$ . Так же у вас есть  $m$  коробок. Размер  $i$ -й коробки  $a_i$ , где  $a_i$  — целая неотрицательная степень двойки.

Вы можете делить коробки на две части одинакового размера. Ваша задача — полностью заполнить сумку.

Например, если  $n = 10$  и  $a = [1, 1, 32]$ , то вам нужно разделить коробку размера 32 на две части размера 16, а затем разделить коробку размера 16. Таким образом, вы сможете заполнить сумку коробками размеров 1, 1 и 8.

Посчитайте минимальное количество делений, которое придется сделать для заполнения сумки размера  $n$ .

### Входные данные

Первая строка содержит число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит два числа  $n$  и  $m$  ( $1 \leq n \leq 10^{18}$ ,  $1 \leq m \leq 10^5$ ) — размер сумки и количество коробок, соответственно.

Вторая строка каждого набора входных данных содержит  $m$  чисел  $a_1, a_2, \dots, a_m$  ( $1 \leq a_i \leq 10^9$ ) — размеры коробок. Гарантируется, что все  $a_i$  — степени двойки.

Также гарантируется, что сумма всех  $m$  по всем наборам входных данных не превосходит  $10^5$ .

### Выходные данные

На каждый набор входных данных выведите число — минимальное количество делений, которые придется сделать для заполнения сумки размера  $n$  (или  $-1$ , если заполнить сумку невозможно).

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 3<br>10 3<br>1 32 1<br>23 4<br>16 1 4 1<br>20 5<br>2 1 16 1 8 |             |
| выходные данные   | Скопировать |
| 2<br>-1<br>0  |             |



## D. Красивый массив

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам дан массив  $a$  состоящий из  $n$  целых чисел. Красота массива — это максимальная сумма какого-то **последовательного подотрезка** этого массива (этот подотрезок может быть пустым). Например красота массива  $[10, -5, 10, -4, 1]$  равна 15, а красота массива  $[-3, -5, -1]$  равна 0.

Вы можете выбрать **не более одного** последовательного подотрезка массива  $a$  и домножить все элементы этого подотрезка на  $x$ . Вы хотите максимизировать красоту массива после применения такой операции.

### Входные данные

Первая строка содержит два целых числа  $n$  и  $x$  ( $1 \leq n \leq 3 \cdot 10^5, -100 \leq x \leq 100$ ) — длина массива  $a$  и число  $x$  соответственно.

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ) — сам массив  $a$ .

### Выходные данные

Выведите одно число — максимальную красоту массива  $a$  после не более одного домножения непрерывного подотрезка этого массива на  $x$ .

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5 -2<br>-3 8 -2 1 -6   |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 22                     |                             |

|                                  |                             |
|----------------------------------|-----------------------------|
| <b>входные данные</b>            | <a href="#">Скопировать</a> |
| 12 -3<br>1 3 3 7 1 3 3 7 1 3 3 7 |                             |
| <b>выходные данные</b>           | <a href="#">Скопировать</a> |
| 42                               |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5 10<br>-1 -2 -3 -4 -5 |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 0                      |                             |

### Примечание

В первом тестовом примере нужно домножить подотрезок  $[-2, 1, -6]$ , после этого массив примет вид  $[-3, 8, 4, -2, 12]$  с красотой 22 ( $[-3, 8, 4, -2, 12]$ ).

Во втором примере нам не нужно домножать какой-либо подотрезок массива вообще.

В третьем тестовом примере независимо от того, какой подотрезок мы домножим, красота массива будет равна 0.





[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

### D. Простая XOR раскраска

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам дан неориентированный граф с  $n$  вершинами, пронумерованными от 1 до  $n$ . Между вершинами  $u$  и  $v$  существует ребро тогда и только тогда, когда  $u \oplus v$  является **простым числом**, где  $\oplus$  обозначает операцию **побитового исключающего ИЛИ**.

Раскрасьте все вершины графа в минимальное количество цветов так, чтобы ни одна пара вершин, непосредственно соединенных ребром, не была покрашена в один цвет.

#### Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 500$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Единственная строка содержит одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество вершин в графе.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

#### Выходные данные

Для каждого набора входных данных выведите две строки.

Первая строка должна содержать одно целое число  $k$  ( $1 \leq k \leq n$ ) — минимально необходимое количество цветов.

Вторая строка должна содержать  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq k$ ) — цвет каждой вершины.

Если существует несколько решений, выведите любое из них.

#### Пример

| входные данные   | Скопировать |
|--|-------------|
| 6<br>1<br>2<br>3<br>4<br>5<br>6  |             |
| выходные данные  | Скопировать |
| 1<br>1<br>2<br>1 2<br>2<br>1 2 2<br>3<br>1 2 2 3<br>3<br>1 2 2 3 3<br>4<br>1 2 2 3 3 4 |             |

#### Примечание

В первом наборе входных данных минимальное количество цветов равно 1, потому что дана только одна вершина.

Во втором наборе входных данных минимальное количество цветов равно 2, потому что существует ребро, соединяющее 1 и 2 ( $1 \oplus 2 = 3$ , что является простым числом).

В третьем наборе входных данных минимальное количество цветов по-прежнему равно 2, потому что 2 и 3 могут быть окрашены одинаково, так как между 2 и 3 нет ребра ( $2 \oplus 3 = 1$ , что не является простым числом).

В четвертом наборе входных данных можно показать, что минимальное количество цветов равно 3.

В пятом наборе входных данных можно показать, что минимальное количество цветов равно 3.

В шестом наборе входных данных можно показать, что минимальное количество цветов равно 4.



## B1. Покраска массива I

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 512 мегабайт

**Единственное различие между двумя версиями задачи заключается в том, что в этой версии вам нужно найти максимальный возможный ответ.**

Гомеру очень нравятся массивы. Сегодня он красит массив  $a_1, a_2, \dots, a_n$  двумя видами цветов, **белым** и **черным**. Покраска массива  $a_1, a_2, \dots, a_n$  описывается массивом  $b_1, b_2, \dots, b_n$ , где  $b_i$  обозначает цвет  $a_i$  (0 для белого и 1 для черного).

Согласно покраске  $b_1, b_2, \dots, b_n$  массив  $a$  разделяется на два новых массива  $a^{(0)}$  и  $a^{(1)}$ , где  $a^{(0)}$  — это подпоследовательность всех белых элементов в  $a$ , а  $a^{(1)}$  — это подпоследовательность всех черных элементов в  $a$ . Например, если  $a = [1, 2, 3, 4, 5, 6]$  и  $b = [0, 1, 0, 1, 0, 0]$ , то  $a^{(0)} = [1, 3, 5, 6]$  и  $a^{(1)} = [2, 4]$ .

Количество отрезков в массиве  $c_1, c_2, \dots, c_k$ , обозначаемое  $seg(c)$ , — это количество элементов, которое останется, если объединить все соседние элементы с одинаковым значением в  $c$ . Например, количество отрезков в  $[1, 1, 2, 2, 3, 3, 3, 2]$  равно 4, так как массив станет равным  $[1, 2, 3, 2]$  после объединения соседних элементов с одинаковым значением. В частности, количество отрезков в пустом массиве равно 0.

Гомер хочет найти покраску  $b$ , согласно которой суммарное количество отрезков в  $a^{(0)}$  и  $a^{(1)}$ , т. е.  $seg(a^{(0)}) + seg(a^{(1)})$ , **максимально**. Найдите, чему равно это значение.

### Входные данные

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ).

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ).

### Выходные данные

Выведите единственное число — **максимально** возможное суммарное количество отрезков.

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 7<br>1 1 2 2 3 3 3     |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 6                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 7<br>1 2 3 4 5 6 7     |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 7                      |                             |

### Примечание

В первом примере можно выбрать  $a^{(0)} = [1, 2, 3, 3]$ ,  $a^{(1)} = [1, 2, 3]$  и  $seg(a^{(0)}) = seg(a^{(1)}) = 3$ . Таким образом, ответ  $3 + 3 = 6$ .

Во втором примере можно выбрать  $a^{(0)} = [1, 2, 3, 4, 5, 6, 7]$  и  $a^{(1)}$  пустое. Мы видим, что  $seg(a^{(0)}) = 7$  и  $seg(a^{(1)}) = 0$ . Таким образом, ответ  $7 + 0 = 7$ .

## Е. Переворот строки

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам задана строка  $s$ . Вам необходимо перевернуть эту строку. То есть последняя буква строки должна стать первой, предпоследняя буква строки должна стать второй, и так далее. Например, перевернутая строка «abddea» равна «aeddba». Для достижения цели (то есть для переворота заданной строки) вы можете менять местами **соседние элементы строки**.

Перед вами стоит задача определить минимальное количество обменов соседних элементов строки, необходимых для того, чтобы перевернуть строку.

### Входные данные

В первой строке следует целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — длина строки  $s$ .

Во второй строке следует строка  $s$  длины  $n$ , состоящая из строчных букв латинского алфавита.

### Выходные данные

Выведите минимальное количество обменов соседних элементов строки, необходимых для того, чтобы перевернуть строку.

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5<br>aaaza             |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 2                      |                             |
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 6<br>cbaabc            |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 0                      |                             |
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 9<br>icpcsguru         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 30                     |                             |

### Примечание

В первом примере нужно сначала поменять местами третью и четвертую буквы, тогда строка станет равна «aazaa». Затем нужно поменять вторую и третью буквы, тогда строка станет равна «azaaa». Таким образом, за два обмена соседних букв мы сможем перевернуть заданную строку.

Во втором примере заданная строка является палиндромом, то есть перевернутая строка равна исходной строке, поэтому никаких обменов делать не нужно.



## F. Сделай связным

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дан неориентированный граф, состоящий из  $n$  вершин. На каждой вершине записано число; число, записанное на вершине  $i$ , равно  $a_i$ . Изначально в графе нет ни одного ребра.

Вы можете добавлять ребра в граф за определенную стоимость. За добавление ребра между вершинами  $x$  и  $y$  надо заплатить  $a_x + a_y$  монет. Также существует  $m$  специальных предложений, каждое из которых характеризуется тремя числами  $x$ ,  $y$  и  $w$ , и означает, что можно добавить ребро между вершинами  $x$  и  $y$  за  $w$  монет. Эти специальные предложения не обязательно использовать: если существует такая пара вершин  $x$  и  $y$ , такая, что для нее существует специальное предложение, можно все равно добавить ребро между ними за  $a_x + a_y$  монет.

Сколько монет минимально вам потребуется, чтобы сделать граф связным? Граф является связным, если от каждой вершины можно добраться до любой другой вершины, используя только ребра этого графа.

### Входные данные

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq n \leq 2 \cdot 10^5$ ,  $0 \leq m \leq 2 \cdot 10^5$ ) — количество вершин в графе и специальных предложений, соответственно.

Во второй строке заданы  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^{12}$ ) — числа, записанные на вершинах.

Затем следуют  $m$  строк, в каждой из которых заданы три целых числа  $x$ ,  $y$  и  $w$  ( $1 \leq x, y \leq n$ ,  $1 \leq w \leq 10^{12}$ ,  $x \neq y$ ), обозначающие спецпредложение: можно добавить ребро между вершинами  $x$  и  $y$  за  $w$  монет.

### Выходные данные

Выведите одно целое число — минимальное количество монет, которое необходимо потратить, чтобы сделать граф связным.

### Примеры

|                                |             |
|--------------------------------|-------------|
| <b>входные данные</b>          | Скопировать |
| 3 2<br>1 3 3<br>2 3 5<br>2 1 1 |             |
| <b>выходные данные</b>         | Скопировать |
| 5                              |             |

|                        |             |
|------------------------|-------------|
| <b>входные данные</b>  | Скопировать |
| 4 0<br>1 3 3 7         |             |
| <b>выходные данные</b> | Скопировать |
| 16                     |             |

|  |             |
|--|-------------|
| <b>входные данные</b>                                  | Скопировать |
| 5 4<br>1 2 3 4 5<br>1 2 8<br>1 3 10<br>1 4 7<br>1 5 15 |             |
| <b>выходные данные</b>                                 | Скопировать |
| 18   |             |

### Примечание

В первом примере из условия можно соединить 1 и 2 при помощи 2-го спецпредложения, а затем 1 и 3 без использования спецпредложения.

В следующих двух примерах оптимальный ответ можно получить без использования спецпредложений.

## D. Чет-нечет подпоследовательность

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

У Ashish есть массив  $a$  размера  $n$ .

Подпоследовательность массива  $a$  определяется, как последовательность, которая может быть получена из  $a$  с помощью удаления некоторых элементов (возможно нуля) без изменения порядка оставшихся элементов.

Рассмотрим подпоследовательность  $s$  массива  $a$ . Он определяет цену подпоследовательности  $s$  как минимум между:

- Максимумом по всем числам, стоящим на нечетных позициях в  $s$ .
- Максимумом по всем числам, стоящим на четных позициях в  $s$ .

Обратите внимание, что индекс числа определяется, как его индекс в  $s$ , независимо от его индекса в  $a$ . Все позиции нумеруются с 1. Другими словами, цена  $s$  равна  $\min(\max(s_1, s_3, s_5, \dots), \max(s_2, s_4, s_6, \dots))$ .

Например, цена последовательности  $\{7, 5, 6\}$  равна  $\min(\max(7, 6), \max(5)) = \min(7, 5) = 5$ .

Помогите ему найти минимальную цену какой-то подпоследовательности размера  $k$ .

### Входные данные

В первой строке находится два целых числа  $n$  и  $k$  ( $2 \leq k \leq n \leq 2 \cdot 10^5$ ) — размер массива  $a$  и размер подпоследовательности.

В следующей строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива  $a$ .

### Выходные данные

Выведите единственное целое число — минимальную цену подпоследовательности размера  $k$ .

### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 4 2<br>1 2 3 4         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 1                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 4 3<br>1 2 3 4         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 2                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5 3<br>5 3 4 2 6       |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 2                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 6 4<br>5 3 50 2 4 5    |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 3                      |                             |

### Примечание

В первом тесте, рассмотрим подпоследовательность  $s = \{1, 3\}$ . Здесь цена подпоследовательности равна  $\min(\max(1), \max(3)) = 1$ .

Во втором тесте, рассмотрим подпоследовательность  $s = \{1, 2, 4\}$ . Здесь цена подпоследовательности равна

## В. Jzzhu и города

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Jzzhu — президент страны A. В его стране есть  $n$  городов, пронумерованных от 1 до  $n$ . Город 1 — столица A. Также есть  $m$  дорог, соединяющих города. По  $i$ -й дороге можно пройти из города  $u_i$  в город  $v_i$  (и наоборот), длина этой дороги равна  $x_i$ . Более того, в стране есть  $k$  железнодорожных маршрутов. По  $i$ -му маршруту можно доехать от столицы страны до города  $s_i$  (и наоборот), длина этого маршрута равна  $y_i$ .

Jzzhu не хочет зря растрачивать государственный бюджет, поэтому он хочет закрыть некоторые железнодорожные маршруты. Пожалуйста, посчитайте для Jzzhu, какое максимальное количество железнодорожных маршрутов можно закрыть, если требуется выполнить следующее условие: длина кратчайшего пути из каждого города в столицу не должна измениться.

### Входные данные

В первой строке записано три целых числа,  $n, m, k$  ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 3 \cdot 10^5$ ;  $1 \leq k \leq 10^5$ ).

В каждой из следующих  $m$  строк записано три целых числа,  $u_i, v_i, x_i$  ( $1 \leq u_i, v_i \leq n$ ;  $u_i \neq v_i$ ;  $1 \leq x_i \leq 10^9$ ).

В каждой из следующих  $k$  строк записано два целых числа,  $s_i$  и  $y_i$  ( $2 \leq s_i \leq n$ ;  $1 \leq y_i \leq 10^9$ ).

Гарантируется, что существует по крайней мере один путь из каждого города в столицу. Обратите внимание, что между двумя городами может быть несколько дорог. Также может быть несколько железнодорожных маршрутов, ведущих из одного и того же города в столицу.

### Выходные данные

Выведите единственное целое число, обозначающее максимальное количество железнодорожных путей, которые можно закрыть.

### Примеры

|   |                             |
|---|-----------------------------|
| <b>входные данные</b>   | <a href="#">Скопировать</a> |
| 5 5 3<br>1 2 1<br>2 3 2<br>1 3 3<br>3 4 4<br>1 5 5<br>3 5<br>4 5<br>5 5 |                             |
| <b>выходные данные</b>  | <a href="#">Скопировать</a> |
| 2   |                             |

|  |                             |
|--|-----------------------------|
| <b>входные данные</b>                        | <a href="#">Скопировать</a> |
| 2 2 3<br>1 2 2<br>2 1 3<br>2 1<br>2 2<br>2 3 |                             |
| <b>выходные данные</b>                       | <a href="#">Скопировать</a> |
| 2  |                             |





ЗАДАЧИ   ОТОСЛАТЬ   СТАТУС   ПОЛОЖЕНИЕ   ЗАПУСК

## В. Орак и медианы

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

У Слайма есть последовательность натуральных чисел  $a_1, a_2, \dots, a_n$ .

За одну операцию Орак может выбрать произвольный отрезок  $[l \dots r]$  этой последовательности и заменить все числа  $a_l, a_{l+1}, \dots, a_r$  на значение медианы  $\{a_l, a_{l+1}, \dots, a_r\}$ .

В этой задаче для мультимножества натуральных чисел  $s$ , медиана  $s$  равна  $\lfloor \frac{|s|+1}{2} \rfloor$ -у числу в порядке возрастания в нем. Например, медиана  $\{1, 4, 4, 6, 5\}$  равна 4, а медиана  $\{1, 7, 5, 8\}$  равна 5.

Слайм хочет, чтобы Орак добился  $a_1 = a_2 = \dots = a_n = k$  с помощью этих операций.

Орак думает, что это невозможно, и он не хочет тратить свое время, поэтому он решил спросить вас, можно ли исполнить желание Слайма, и он может задавать вам запросы такого вида несколько раз.

### Входные данные

В первой строке записано одно целое число  $t$ : количество запросов.

В первой строке каждого запроса записано два целых числа  $n$  ( $1 \leq n \leq 100\,000$ ) и  $k$  ( $1 \leq k \leq 10^9$ ), во второй строке записаны  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )

Сумма  $n$  по всем запросам не превосходит 100 000.

### Выходные данные

Вы должны вывести  $t$  строк. В  $i$ -й из них должно быть записано 'yes', если возможно превратить все числа в  $k$  за какое-нибудь число операций или 'no', иначе. Каждая выведенная буква может быть как строчной, так и заглавной.

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 5<br>5 3<br>1 5 2 6 1<br>1 6<br>6<br>3 2<br>1 2 3<br>4 3<br>3 1 2 3<br>10 3<br>1 2 3 4 5 6 7 8 9 10 |             |
| выходные данные   | Скопировать |
| no<br>yes<br>yes<br>no<br>yes   |             |

### Примечание

В первом запросе Орак не может превратить все элементы в 3.

Во втором запросе  $a_1 = 6$  уже выполнено.

В третьем запросе Орак может для операции выбрать весь массив и превратить его в 2.

В четвертом запросе Орак не может превратить все элементы в 3.

В пятом запросе Орак может сначала выбрать  $[1, 6]$ , а затем  $[2, 10]$ .

ЗАДАЧИ   ОТОСЛАТЬ   СТАТУС   ПОЛОЖЕНИЕ   ЗАПУСК

## Е. Достижимость из столицы

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

В Берляндии  $n$  городов и  $m$  дорог, каждая из дорог соединяет пару городов. Дороги в Берляндии — односторонние.

Какое наименьшее количество новых дорог надо построить, чтобы из столицы Берляндии по дорогам стали достижимы все города страны?

Построенные новые дороги тоже будут односторонними.

### Входные данные

В первой строке задано три целых числа  $n$ ,  $m$  и  $s$  ( $1 \leq n \leq 5000$ ,  $0 \leq m \leq 5000$ ,  $1 \leq s \leq n$ ) — количество городов, количество дорог и номер столицы. Города пронумерованы от 1 до  $n$ .

В следующих  $m$  строках заданы дороги:  $i$ -я дорога задается парой номеров соединяемых городов  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ). Для каждой пары городов  $(u, v)$  может быть не более одной дороги из  $u$  в  $v$ , но существование встречных дорог (из  $u$  в  $v$  и одновременно из  $v$  в  $u$ ) допустимо. Все дороги в Берляндии — односторонние.

### Выходные данные

Выведите одно целое число — минимальное количество дорог, которое необходимо построить, чтобы из города  $s$  стали достижимы все остальные города. Если во входных данных уже все города достижимы из  $s$ , то выведите 0.

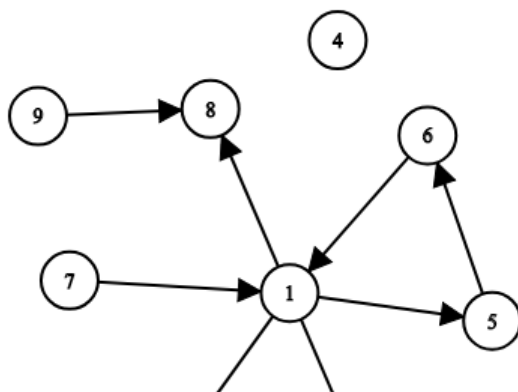
### Примеры

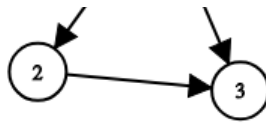
| входные данные   | Скопировать |
|--|-------------|
| 9 9 1<br>1 2<br>1 3<br>2 3<br>1 5<br>5 6<br>6 1<br>1 8<br>9 8<br>7 1 |             |
| выходные данные  | Скопировать |
| 3  |             |

| входные данные                    | Скопировать |
|-----------------------------------|-------------|
| 5 4 5<br>1 2<br>2 3<br>3 4<br>4 1 |             |
| выходные данные                   | Скопировать |
| 1                                 |             |

### Примечание

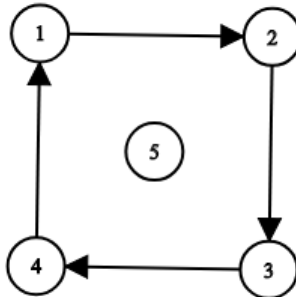
Первый пример изображен на следующем рисунке:





Для того, чтобы все города стали достижимы из  $s = 1$ , можно добавить, например, дороги  $(6, 4)$ ,  $(7, 9)$ ,  $(1, 7)$ .

Второй пример:



В этом примере можно добавить любую из дорог  $(5, 1)$ ,  $(5, 2)$ ,  $(5, 3)$ ,  $(5, 4)$ , чтобы все остальные города стали достижимы из  $s = 5$ .

---

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:40:10<sup>UTC+5</sup> (i1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



**ИТМО**



## C. Очередная задача про массив

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дан массив  $a$  из  $n$  целых чисел. Вы можете совершить следующую операцию любое число раз (0 или больше раз):

- Выбрать 2 индекса  $i, j$  где  $1 \leq i < j \leq n$  и заменить  $a_k$  для всех  $i \leq k \leq j$  значением  $|a_i - a_j|$

Выведите максимальную сумму всех элементов конечного массива, которую вы можете получить таким образом.

### Входные данные

В первой строке содержится одно целое число  $t$  ( $1 \leq t \leq 10^5$ ) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — длина массива  $a$ .

Вторая строка каждого набора содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — элементы массива  $a$ .

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

### Выходные данные

Для каждого набора входных данных выведите сумму конечного массива.

### Пример

| входные данные                            | Скопировать |
|---|-------------|
| 3<br>3<br>1 1 1<br>2<br>9 1<br>3<br>4 9 5 |             |
| выходные данные                           | Скопировать |
| 3<br>16<br>18                             |             |

### Примечание

В первом примере невозможно достичь сумму  $> 3$  используя операцию, таким образом ответ 3.

Во втором примере можно показать, что максимальная достижимая сумма равна 16. Используем операцию (1, 2) и преврати массив из  $[9, 1]$  в  $[8, 8]$ , таким образом ответ равен 16.

В третьем примере можно показать, что невозможно достичь суммы  $> 18$  используя операцию, таким образом ответ 18.



## Е. Перестановка дерева

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

У Ashish есть дерево, состоящее из  $n$  вершин, пронумерованных от 1 до  $n$ , с корнем в вершине 1.  $i$ -я вершина дерева имеет стоимость  $a_i$ , и в ней записана бинарная цифра  $b_i$ . Ashish хочет, чтобы в конце в  $i$ -й вершине была записана бинарная цифра  $c_i$ .

Для этого, он может выполнить следующую операцию любое количество раз:

- Выберите любые  $k$  вершин из поддерева любой вершины  $u$  и переставьте цифры в этих вершинах так, как пожелаете, за стоимость  $k \cdot a_u$ . Здесь он может выбрать  $k$  в диапазоне от 1 до размера поддерева  $u$ .

Он хочет выполнить операции так, чтобы у каждой вершины в итоге оказалась цифра, соответствующая желаемой цифре для этой вершины.

Помогите ему найти минимальную общую стоимость, за которую можно сделать так, чтобы после проведения всех операций для каждого  $u$  в вершине  $u$  была записана цифра  $c_u$ , или определить, что это невозможно.

### Входные данные

Первая строка содержит одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ), обозначающее количество вершин в дереве.

$i$ -я из следующих  $n$  строк содержит 3 разделенных пробелом целых числа  $a_i$ ,  $b_i$ ,  $c_i$  ( $1 \leq a_i \leq 10^9$ ,  $0 \leq b_i, c_i \leq 1$ ) — стоимость  $i$ -й вершины, ее начальная цифра и желаемая цифра.

Каждая из следующих строк  $n - 1$  содержит два целых числа  $u, v$  ( $1 \leq u, v \leq n$ ,  $u \neq v$ ), что означает, что между вершинами  $u$  и  $v$  в дереве есть ребро.

### Выходные данные

Выведите минимальную общую стоимость, за которую можно сделать так, чтобы в каждой вершине была записана желаемая цифра для этой вершины, или  $-1$ , если сделать это невозможно.

### Примеры

|  |                    |
|--|--------------------|
| <b>входные данные</b>  | <b>Скопировать</b> |
| 5<br>1 0 1<br>20 1 0<br>300 0 1<br>4000 0 0<br>50000 1 0<br>1 2<br>2 3<br>2 4<br>1 5 |                    |
| <b>выходные данные</b>   | <b>Скопировать</b> |
| 4  |                    |

|  |                    |
|--|--------------------|
| <b>входные данные</b>  | <b>Скопировать</b> |
| 5<br>10000 0 1<br>2000 1 0<br>300 0 1<br>40 0 0<br>1 1 0<br>1 2<br>2 3<br>2 4<br>1 5 |                    |
| <b>выходные данные</b>   | <b>Скопировать</b> |
| 24000  |                    |

|                         |                    |
|-------------------------|--------------------|
| <b>входные данные</b>   | <b>Скопировать</b> |
| 2<br>109 0 1<br>205 0 1 |                    |

1 2

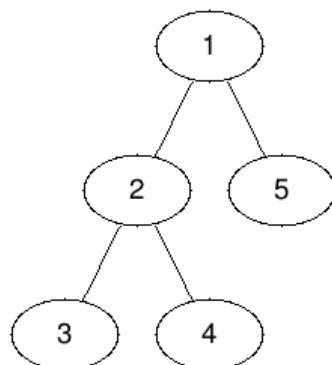
выходные данные

Скопировать

-1

**Примечание**

Дерево, соответствующие примерам 1 и 2:



В примере 1 мы можем выбрать вершину 1 и  $k = 4$  за стоимость  $4 \times 1 = 4$  и выбрать вершины 1, 2, 3, 5, переставить их цифры и получить желаемые цифры в каждой позиции.

В примере 2 мы можем выбрать вершину 1 и  $k = 2$  за стоимость  $10000 \times 2$ , выбрать вершины 1, 5, обменять их цифры, после чего аналогичным образом выбрать вершину 2 и  $k = 2$  за стоимость  $2000 \times 2$  и вершины 2, 3, обменять их цифры, чтобы получить нужные цифры в каждой позиции.

В примере 3 невозможно получить нужные цифры, потому что среди начальных цифр нет цифры 1.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 17.07.2025 12:40:13<sup>UTC+5</sup> (i1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке

**ИТМО**



ЗАДАЧИ   ОТОСЛАТЬ   СТАТУС   ПОЛОЖЕНИЕ   ЗАПУСК

## Е. Уничтожение стены

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Монокарп играет в стратегическую компьютерную игру Rage of Empires II: Definitive Edition. Сейчас он собирается атаковать своего оппонента в игре, но вот незадача — противник успел построить стену, и войска Монокарпа не могут попасть в лагерь противника!

Стена состоит из  $n$  секций, расположенных в ряд и пронумерованных слева направо, начиная с единицы. Прочность  $i$ -й секции изначально равна  $a_i$ . Если прочность какой-то секции станет равной 0 или станет отрицательной, то эта секция будет считаться разрушенной.

Для успешной атаки Монокарп должен разрушить хотя бы две секции стены (любые две: возможно, соседние, но не обязательно). Для этого собирается использовать специальное осадное орудие — онагр. Когда онагр стреляет по секции стены, он наносит 2 единицы урона этой секции и по 1 единице урона соседним секциям. Иными словами, если онагр стреляет по секции  $x$ , прочность секции  $x$  уменьшается на 2, а прочности секций  $x - 1$  и  $x + 1$  (если они существуют) уменьшаются на 1 каждая.

Монокарп может стрелять из онагра по любым секциям любое количество раз, **в том числе можно стрелять по уже разрушенной секции**.

Монокарп хочет посчитать минимальное количество выстрелов онагра, необходимое, чтобы разрушить хотя бы две секции стены. Помогите ему!

### Входные данные

В первой строке следует целое число  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — количество секций в стене.

Во второй строке следует последовательность целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6$ ), где  $a_i$  равно изначальной прочности  $i$ -й секции стены.

### Выходные данные

Выведите целое число — минимальное количество выстрелов онагра, необходимое, чтобы разрушить хотя бы две секции стены.

### Примеры

|                     |             |
|---------------------|-------------|
| входные данные      | Скопировать |
| 5<br>20 10 30 10 20 |             |
| выходные данные     | Скопировать |
| 10                  |             |
| входные данные      | Скопировать |
| 3<br>1 8 1          |             |
| выходные данные     | Скопировать |
| 1                   |             |
| входные данные      | Скопировать |
| 6<br>7 6 6 8 5 8    |             |
| выходные данные     | Скопировать |
| 4                   |             |
| входные данные      | Скопировать |
| 6<br>14 3 8 10 15 4 |             |
| выходные данные     | Скопировать |
| 4                   |             |
| входные данные      | Скопировать |

|                 |             |
|-----------------|-------------|
| 4               |             |
| 1 100 100 1     |             |
| выходные данные | Скопировать |
| 2               |             |

|                 |             |
|-----------------|-------------|
| входные данные  | Скопировать |
| 3               |             |
| 40 10 10        |             |
| выходные данные | Скопировать |
| 7               |             |

Примечание

В первом примере можно разрушить 2-ю и 4-ю секции стены за 10 выстрелов, например, совершив все 10 выстрелов в третью секцию. После этого прочности секций будут равны [20, 0, 10, 0, 20]. Существует и другой способ. Можно сначала выстрелить 5 раз во 2-ю секцию и затем еще 5 раз в 4-ю секцию. После этого прочности секций будут равны [15, 0, 20, 0, 15].

Во втором примере достаточно одного выстрела по 2-й секции. После этого 1-я и 3-я секции будут разрушены.

В третьем примере можно, например, выстрелить два раза во 2-ю секцию (после этого прочности секций станут равны [5, 2, 4, 8, 5, 8]), а затем выстрелить два раза в 3-ю секцию (после этого прочности секций станут равны [5, 0, 0, 6, 5, 8]). Таким образом, после четырех выстрелов будут разрушены 2-я и 3-я секции.

Codeforces (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:40:16<sup>UTC+5</sup> (i1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



## D. Капитан Флинт и сокровище

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В погоне за очередным сокровищем капитан Флинт наткнулся на некоторую задачу, которая может быть связана с поиском сокровища, а может и не быть. Поэтому капитан Флинт поручил ее решение своим матросам и назначил огромное вознаграждение: целый один выходной день. Задача же звучит так...

Заданы массивы  $a$  и  $b$  длины  $n$ . Изначально,  $ans$  равен 0 и определена следующая операция:

1. Выбрать позицию  $i$  ( $1 \leq i \leq n$ );
2. Добавить к  $ans$  значение  $a_i$ ;
3. Если  $b_i \neq -1$ , то добавить к  $a_{b_i}$  значение  $a_i$ ;

Какой максимальный  $ans$  можно получить, применив эту операцию к каждой позиции  $i$  ( $1 \leq i \leq n$ ) ровно один раз?

Дядя Богдан очень хочет получить вознаграждение и просит Вас помочь ему с решением задачи, а также предоставить порядок позиций, в котором нужно применять операцию выше.

### Входные данные

В первой строке каждого набора задано одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — длина массивов  $a$  и  $b$ .

Во второй строке каждого набора задано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^6 \leq a_i \leq 10^6$ ).

В третьей строке каждого набора задано  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq n$  или  $b_i = -1$ ).

**Дополнительное ограничение:** гарантируется, что для  $i$  ( $1 \leq i \leq n$ ) последовательность  $b_i, b_{b_i}, b_{b_{b_i}}, \dots$  не заикнется, то есть будет всегда оканчиваться на  $-1$ .

### Выходные данные

В первой строке выведите одно целое число, максимальный  $ans$ , который можно получить.

Во второй строке опишите последовательность выполнения операций, чтобы получить этот максимальный ответ:  $n$  различных целых чисел  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  обозначает позицию, операция над которой выполняется  $i$ -й по порядку. Если существует несколько таких последовательностей, выведите любую из них.

### Примеры

|  |                             |
|--|-----------------------------|
| <b>входные данные</b>                                    | <a href="#">Скопировать</a> |
| 3<br>1 2 3<br>2 3 -1                                     |                             |
| <b>выходные данные</b>                                   | <a href="#">Скопировать</a> |
| 10<br>1 2 3  |                             |
| <b>входные данные</b>                                    | <a href="#">Скопировать</a> |
| 2<br>-1 100<br>2 -1                                      |                             |
| <b>выходные данные</b>                                   | <a href="#">Скопировать</a> |
| 99<br>2 1  |                             |
| <b>входные данные</b>                                    | <a href="#">Скопировать</a> |
| 10<br>-10 -1 2 2 5 -2 -3 -4 2 -6<br>-1 -1 2 2 -1 5 5 7 9 |                             |
| <b>выходные данные</b>                                   | <a href="#">Скопировать</a> |
| -9<br>3 5 6 1 9 4 10 7 8 2                               |                             |

## В. Приравнять их всех

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт

Дан массив  $a$ , состоящий из  $n$  **положительных** целых чисел, пронумерованных от 1 до  $n$ . Вы можете выполнять с этим массивом следующую операцию не более  $3n$  раз:

- выбрать три целых числа  $i, j$  и  $x$  ( $1 \leq i, j \leq n; 0 \leq x \leq 10^9$ );
- присвоить  $a_i := a_i - x \cdot i, a_j := a_j + x \cdot i$ .

После каждой операции все элементы должны быть **неотрицательными**.

Можете ли вы задать такую последовательность из не более чем  $3n$  операций, после которой все элементы массива будут равны?

### Входные данные

Первая строка теста содержит одно целое число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных. Затем следует  $t$  наборов входных данных.

Первая строка набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 10^4$ ) — длина массива. Вторая строка набора содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^5$ ) — элементы массива.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $10^4$ .

### Выходные данные

Для каждого набора входных данных выведите ответ следующим образом:

- если описанной в условии последовательности операций не существует, выведите  $-1$ ;
- иначе выведите целое число  $k$  ( $0 \leq k \leq 3n$ ) — количество операций, которые вы совершаете. Затем выведите  $k$  строк,  $m$ -я из которых должна содержать три целых числа  $i, j$  и  $x$  ( $1 \leq i, j \leq n; 0 \leq x \leq 10^9$ ) для  $m$ -й операции.

Если возможных последовательностей операций несколько, выведите любую из них. Обратите внимание, что минимизировать  $k$  не обязательно.

### Пример

| входные данные   | Скопировать |
|--|-------------|
| 3<br>4<br>2 16 4 18<br>6<br>1 2 3 4 5 6<br>5<br>11 19 1 1 3        |             |
| выходные данные  | Скопировать |
| 2<br>4 1 2<br>2 3 3<br>-1<br>4<br>1 2 4<br>2 4 5<br>2 3 3<br>4 5 1 |             |

### С. Соединение точек

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам дан набор точек  $x_1, x_2, \dots, x_n$  на числовой прямой.

Две точки  $i$  и  $j$  могут быть соединены друг с другом при выполнении следующих условий:

- Ни  $i$ , ни  $j$  не соединены ни с какими другими точками;
- $|x_i - x_j| \geq z$ .

Какое максимальное количество пар точек вы можете соединить друг с другом?

#### Входные данные

Первая строка содержит два целых числа  $n$  и  $z$  ( $2 \leq n \leq 2 \cdot 10^5, 1 \leq z \leq 10^9$ ) — количество точек и ограничение на дистанцию между соединяемыми точками, соответственно.

Вторая строка содержит  $n$  целых чисел  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 10^9$ ).

#### Выходные данные

Выведите одно целое число — максимальное количество пар точек, которое вы можете соединить друг с другом.

#### Примеры

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 4 2<br>1 3 3 7         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 2                      |                             |

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5 5<br>10 9 5 8 7      |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 1                      |                             |

#### Примечание

В первом примере вы можете соединить точку 1 с точкой 2 ( $|3 - 1| \geq 2$ ), и точку 3 с точкой 4 ( $|7 - 3| \geq 2$ ).

Во втором примере вы можете соединить точку 1 с точкой 3 ( $|5 - 10| \geq 5$ ).





[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

## F. Три пути в дереве

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам задано невзвешенное дерево с  $n$  вершинами. Напомним, что дерево — это связный неориентированный граф без циклов.

Ваша задача — выбрать **три различных** вершины  $a, b, c$  в этом дереве таких, что количество ребер, принадлежащих **как минимум** одному из простых путей между  $a$  и  $b$ ,  $b$  и  $c$ , или  $a$  и  $c$ , максимально. Обратите внимание на примечания для лучшего понимания.

Простой путь — это путь, который посещает каждую вершину не более одного раза.

### Входные данные

Первая строка содержит единственное целое число  $n$  ( $3 \leq n \leq 2 \cdot 10^5$ ) — количество вершин в дереве.

Следующая  $n - 1$  строка описывает ребра дерева в виде  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ). Гарантируется, что заданный граф является деревом.

### Выходные данные

Первой строкой выведите одно целое число  $res$  — максимальное количество ребер, принадлежащих **как минимум** одному из простых путей между  $a$  и  $b$ ,  $b$  и  $c$ , или  $a$  и  $c$ .

Во второй строке выведите три целых числа  $a, b, c$  таких, что  $1 \leq a, b, c \leq n$  и  $a \neq b, b \neq c, a \neq c$ .

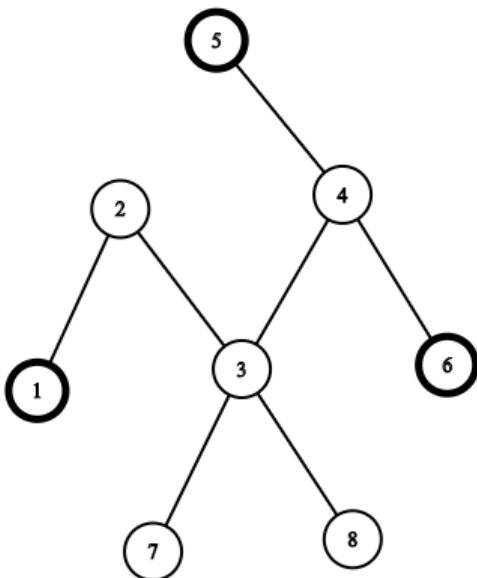
Если существует несколько подходящих ответов, вы можете вывести любой.

### Пример

| входные данные                                     | Скопировать |
|--|-------------|
| 8<br>1 2<br>2 3<br>3 4<br>4 5<br>4 6<br>3 7<br>3 8 |             |
| выходные данные                                    | Скопировать |
| 5<br>1 8 6   |             |

### Примечание

Изображение, соответствующее первому примеру (и **другой правильный ответ**):



## В. Домино для молодых

ограничение по времени на тест: 3 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам дана диаграмма Юнга.

Данная диаграмма это гистограмма с  $n$  столбцами длин  $a_1, a_2, \dots, a_n$  ( $a_1 \geq a_2 \geq \dots \geq a_n \geq 1$ ).

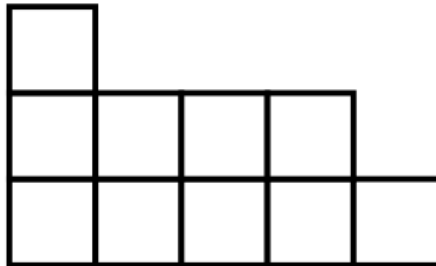


Диаграмма Юнга для  $a = [3, 2, 2, 2, 1]$ .

Ваша задача — найти наибольшее количество непересекающихся домино, которое вы можете нарисовать внутри этой диаграммы, домино это прямоугольник  $1 \times 2$  или  $2 \times 1$ .

### Входные данные

В первой строке записано одно целое число  $n$  ( $1 \leq n \leq 300\,000$ ): количество столбцов в данной гистограмме.

В следующей строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 300\,000, a_i \geq a_{i+1}$ ): длины столбцов.

### Выходные данные

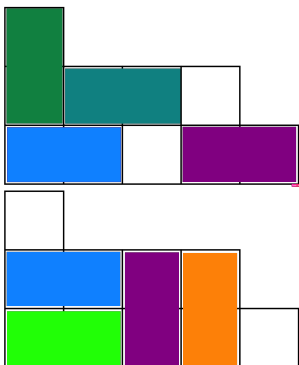
Выведите одно целое число: наибольшее количество непересекающихся домино, которое вы можете нарисовать внутри данной диаграммы Юнга.

### Пример

|                        |                             |
|------------------------|-----------------------------|
| <b>входные данные</b>  | <a href="#">Скопировать</a> |
| 5<br>3 2 2 2 1         |                             |
| <b>выходные данные</b> | <a href="#">Скопировать</a> |
| 4                      |                             |

### Примечание

Некоторые возможные решения для первого примера:



## E1. Деление весов (простая версия)

ограничение по времени на тест: 3 секунды  
ограничение по памяти на тест: 256 мегабайт

**Простая и сложная версии на самом деле являются разными задачами, поэтому прочитайте условия обеих задач внимательно.**

Вам задано взвешенное корневое дерево, вершина 1 — корень этого дерева.

Дерево — это связный граф без циклов. У корневого дерева есть специальная вершина, называемая корнем. Предком вершины  $v$  называется последняя отличная от  $v$  вершина на пути от корня к вершине  $v$ . Потомками вершины  $v$  называются все вершины, для которых  $v$  является предком. Вершина называется листом, если у нее нет потомков. Взвешенное дерево — это такое дерево, в котором у каждого ребра есть некоторый вес.

Вес пути — это сумма весов всех ребер на этом пути. Вес пути от вершины до самой себя равен 0.

Вы можете совершить последовательность из нуля или более ходов. В течение одного хода вы можете выбрать ребро и поделить его вес на 2 с округлением вниз. Более формально, в течение одного хода вы выбираете какое-то ребро  $i$  и делите его вес на 2 с округлением вниз ( $w_i := \lfloor \frac{w_i}{2} \rfloor$ ).

Ваша задача — найти минимальное количество **ходов**, необходимых для того, чтобы сделать **сумму весов путей** от корня до каждого листа не превосходящей  $S$ . Другими словами, если  $w(i, j)$  — это вес пути от вершины  $i$  до вершины  $j$ , то вам необходимо сделать  $\sum_{v \in \text{leaves}} w(\text{root}, v) \leq S$ , где  $\text{leaves}$  — это список всех листьев.

Вам необходимо ответить на  $t$  независимых наборов тестовых данных.

### Входные данные

Первая строка теста содержит одно целое число  $t$  ( $1 \leq t \leq 2 \cdot 10^4$ ) — количество наборов тестовых данных. Затем следуют  $t$  наборов тестовых данных.

Первая строка набора тестовых данных содержит два целых числа  $n$  и  $S$  ( $2 \leq n \leq 10^5$ ;  $1 \leq S \leq 10^{16}$ ) — количество вершин в дереве и максимально возможная сумма весов, которую вы можете получить. Следующая  $n - 1$  строка описывает ребра дерева. Ребро  $i$  описывается тремя целыми числами  $v_i$ ,  $u_i$  и  $w_i$  ( $1 \leq v_i, u_i \leq n$ ;  $1 \leq w_i \leq 10^6$ ), где  $v_i$  и  $u_i$  — вершины, которые соединены ребром  $i$ , а  $w_i$  — вес этого ребра.

Гарантируется, что сумма всех  $n$  не превосходит  $10^5$  ( $\sum n \leq 10^5$ ).

### Выходные данные

Для каждого набора тестовых данных выведите ответ на него: минимальное количество **ходов**, необходимое, чтобы сделать сумму весов путей от корня до каждого листа не превосходящей  $S$ .

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 3<br>3 20<br>2 1 8<br>3 1 7<br>5 50<br>1 3 100<br>1 5 10<br>2 3 123<br>5 4 55<br>2 100<br>1 2 409 |             |
| выходные данные   | Скопировать |
| 0<br>8<br>3   |             |

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

## D. НОД и МОД

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Вам дан массив  $a$  из  $n$  ( $n \geq 2$ ) положительных целых чисел, а также целое число  $p$ . Рассмотрим неориентированный взвешенный граф на  $n$  вершинах, пронумерованных от 1 до  $n$ , в котором между вершинами  $i$  и  $j$  ( $i < j$ ) добавлены следующие ребра:

- Если  $\gcd(a_i, a_{i+1}, a_{i+2}, \dots, a_j) = \min(a_i, a_{i+1}, a_{i+2}, \dots, a_j)$ , то между вершинами  $i$  и  $j$  существует ребро веса  $\min(a_i, a_{i+1}, a_{i+2}, \dots, a_j)$ .
- Если  $i + 1 = j$ , то между вершинами  $i$  и  $j$  существует ребро веса  $p$ .

Здесь  $\gcd(x, y, \dots)$  обозначает **наибольший общий делитель (НОД)** чисел  $x, y, \dots$

Обратите внимание, между вершинами  $i$  и  $j$  появляются кратные ребра, если оба условия выполнены. Если же ни одно условие не выполнено для  $i$  и  $j$ , то между ними нет ребер.

Ваша цель — найти вес **минимального остовного дерева** данного графа.

### Входные данные

Первая строка содержит одно целое число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа  $n$  и  $p$  ( $2 \leq n \leq 2 \cdot 10^5$ ,  $1 \leq p \leq 10^9$ ) — количество вершин и параметр  $p$ .

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, a_3, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

### Выходные данные

Выведите  $t$  строк. Для каждого набора входных данных выведите вес соответствующего минимального остовного дерева.

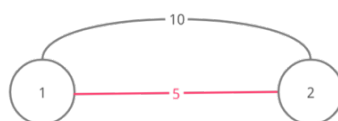
### Пример

| входные данные  | Скопировать |
|---|-------------|
| 4<br>2 5<br>10 10<br>2 5<br>3 3<br>4 5<br>5 2 4 9<br>8 8<br>5 3 3 6 10 100 9 15 |             |
| выходные данные   | Скопировать |
| 5<br>3<br>12<br>46  |             |

### Примечание

Ниже изображены графы для четырех наборов входных данных примера (ребра одного из минимальных остовных деревьев показаны розовым):

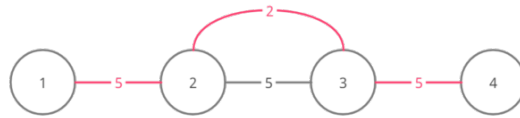
#### Набор 1



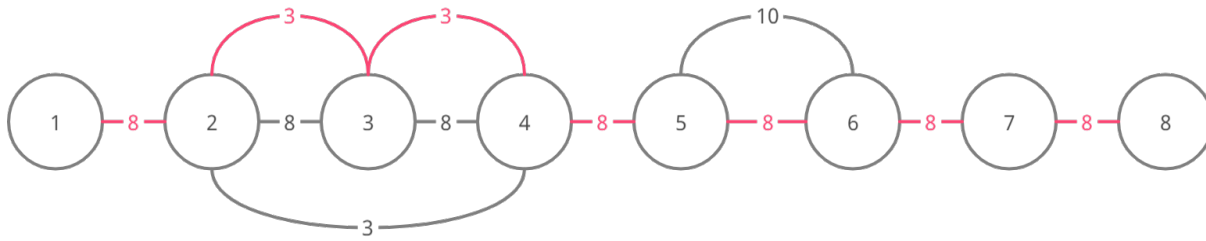
#### Набор 2



## Набор 3



## Набор 4



---

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:40:29<sup>UTC+5</sup> (i1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

## D. Покупка драгоценностей

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт

Nightwish feat. Jonsu - Erämaan Viimeinen



У Алисы есть  $n$  монет, и она хочет сделать покупки в ювелирном магазине Боба. Хотя Боб еще не открыл магазин, он хочет быть уверен, что Алиса сегодня купит **ровно**  $k$  драгоценностей. Для этого Боб может построить не более 60 прилавков (каждый из которых содержит неограниченное количество драгоценностей) и установить цену за драгоценность для каждого прилавка как целое число монет от 1 до  $10^{18}$ .

К счастью, Боб знает, что Алиса покупает жадно: она подойдет к прилавку 1, купит наибольшее возможное количество драгоценностей, затем перейдет к прилавку 2, купит наибольшее возможное количество драгоценностей, и так до последнего прилавка. Зная это, Боб может выбрать количество прилавков, а также установить цену для каждого прилавка так, чтобы Алиса купила ровно  $k$  драгоценностей. Помогите Бобу решить эту задачу или определите, что это невозможно.

Обратите внимание, не обязательно, чтобы Алиса потратила все монеты.

### Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Каждый набор входных данных содержит два целых положительных числа  $n$  и  $k$  ( $1 \leq n, k \leq 10^{18}$ ) — количество монет у Алисы и количество драгоценностей, которое Боб хочет, чтобы Алиса купила.

### Выходные данные

Для каждого набора входных данных выведите в отдельной строке «YES», если Боб может построить не более 60 прилавков и установить цены в них так, чтобы Алиса купила ровно  $k$  драгоценностей, или «NO», если это невозможно.

Если ответ «YES», во второй строке выведите целое число  $s$  ( $1 \leq s \leq 60$ ) — количество прилавков, которое должен построить Боб. В третьей строке выведите  $s$  целых положительных чисел  $p_1, p_2, \dots, p_s$  ( $1 \leq p_i \leq 10^{18}$ ), которые представляют собой подходящий набор цен  $p$ , где  $p_i$  — цена за драгоценный камень на прилавке  $i$ . Если таких  $p$  несколько, выведите любое из них.

### Пример

| входные данные   | Скопировать |
|--|-------------|
| 3<br>7 3<br>6 4<br>255 8   |             |
| выходные данные  | Скопировать |
| YES<br>10<br>2 3 4 5 6 7 8 9 10 11<br>NO<br>YES<br>8<br>128 64 32 16 8 4 2 1 |             |

**Примечание**

В первом наборе входных данных, на первом прилавке Алиса покупает 3 драгоценности и остается с 1 монетой. Этого недостаточно, чтобы купить драгоценности на любом из оставшихся прилавков, поэтому в итоге Алиса купит ровно 3 драгоценности.

В третьем наборе входных данных,

- На первом прилавке Алиса покупает 1 драгоценность и остается с 127 монетами.
- На втором прилавке Алиса покупает 1 драгоценность и остается с 63 монетами.
- На третьем прилавке Алиса покупает 1 драгоценность и остается с 31 монетой.
- На четвертом прилавке Алиса покупает 1 драгоценность и остается с 15 монетами.
- На пятом прилавке Алиса покупает 1 драгоценность и остается с 7 монетами.
- На шестом прилавке Алиса покупает 1 драгоценность и остается с 3 монетами.
- На седьмом прилавке Алиса покупает 1 драгоценность и остается с 1 монетой.
- На восьмом прилавке Алиса покупает 1 драгоценность и остается с 0 монетами.

Таким образом, всего Алиса купит ровно 8 драгоценностей.

---

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов  
Соревнования по программированию 2.0  
Время на сервере: 17.07.2025 12:40:30<sup>UTC+5</sup> (j1).  
Мобильная версия, переключиться на [десктопную](#).  
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке

**ИТМО**

## С. Выбор цветов для букета

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 512 мегабайт

Володя хочет красиво поздравить свою жену с годовщиной их свадьбы и собирается подарить ей букет из **ровно**  $n$  цветов.

Придя в цветочный магазин, Володя обнаружил, что букет можно составлять из цветов  $m$  различных видов, причём количество цветов каждого вида не ограничено. Володя знает, что от первого цветка  $i$ -го вида в букете его супруга становится счастливее на  $a_i$ , а от каждого следующего цветка такого вида она становится счастливее на  $b_i$ . То есть, если в букете  $x_i > 0$  цветов вида  $i$ , то за цветы этого вида жена Володи становится счастливее на  $a_i + (x_i - 1) \cdot b_i$  (а в случае, если цветов типа  $i$  в букете не будет, счастье жены Володи из-за цветов этого типа не изменится).

Как любой любящий муж, Володя хочет как можно сильнее порадовать свою супругу. Поэтому ему хочется знать, на какую максимальную величину может увеличиться её счастье от букета из  $n$  цветов, набранного из доступных в магазине цветов.

### Входные данные

В первой строке задано одно целое число  $t$  ( $1 \leq t \leq 10\,000$ ) — количество наборов входных данных. Затем следует описание  $t$  наборов входных данных, каждый из которых задан следующим образом.

В первой строке набора записаны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^9$ ,  $1 \leq m \leq 100\,000$ ) — требуемое количество цветов в букете и количество доступных видов цветов.

Каждая из следующих  $m$  строк описывает вид цветов и содержит два целых числа  $a_i$  и  $b_i$  ( $0 \leq a_i, b_i \leq 10^9$ ) — увеличение счастья от первого цветка  $i$ -го вида и увеличение счастья от каждого последующего цветка этого вида.

Наборы разделены одной пустой строкой.

Гарантируется, что сумма  $m$  по всем наборам входных данных не превосходит 100 000.

### Выходные данные

Для каждого набора входных данных в единственной строке выведите одно число — максимальное увеличение счастья, которое может получить жена Володи от букета из  $n$  цветов.

### Пример

| входные данные  | Скопировать |
|---|-------------|
| 2<br>4 3<br>5 0<br>1 4<br>2 2<br><br>5 3<br>5 2<br>4 2<br>3 1 |             |
| выходные данные   | Скопировать |
| 14<br>16  |             |

### Примечание

В первом примере если взять 1 цветок первого типа и 3 цветка второго типа, то итоговое увеличение счастья от букета будет равно  $5 + (1 + 2 \cdot 4) = 14$ .

В втором примере если взять 2 цветка первого типа, 2 цветка второго типа и 1 цветок третьего типа, то итоговое увеличение счастья от букета будет равно  $(5 + 1 \cdot 2) + (4 + 1 \cdot 2) + 3 = 16$ .