

А. Прибавление степеней

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

У вас есть массив a длины n . Для каждого положительного числа x в течение x -й секунды вы собираетесь выполнить следующую операцию:

- Выберите несколько различных индексов i_1, i_2, \dots, i_k , которые находятся в диапазоне от 1 до n включительно, и добавьте 2^{x-1} к каждой соответствующей позиции в a . Формально $a_{i_j} := a_{i_j} + 2^{x-1}$ для $j = 1, 2, \dots, k$. **Обратите внимание, что вы можете также не выбрать ни одного индекса.**

Вы должны сделать a неубывающим как можно быстрее. Найдите наименьшее число T такое, что вы можете сделать массив неубывающим не позднее, чем через T секунд.

Массив a называется неубывающим, если и только если $a_1 \leq a_2 \leq \dots \leq a_n$.

Вы должны ответить на t независимых тестовых случаев.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество тестовых случаев.

Первая строка каждого тестового случая содержит одно целое число n ($1 \leq n \leq 10^5$) — длину массива a . Гарантируется, что сумма значений n по всем тестовым случаям не превышает 10^5 .

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

Выходные данные

Для каждого тестового примера выведите минимальное количество секунд, за которое вы можете сделать a неубывающим.

Пример

входные данные	Скопировать
3 4 1 7 6 5 5 1 2 3 4 5 2 0 -4	
выходные данные	Скопировать
2 0 3	

Примечание

В первом тестовом случае, если вы выберете индексы 3, 4 на 1-й секунде и 4 на 2-й секунде, то a станет равным $[1, 7, 7, 8]$. Есть и другие способы сделать a неубывающим за 2 секунды, но вы не сделать его неубывающим быстрее.

Во втором тестовом случае a уже неубывающий, поэтому ответ равен 0.

В третьем тестовом случае, если вы ничего не сделаете в первые 2 секунды и в течение 3-й секунды выберете индекс 2, a станет равным $[0, 0]$.



C1. Настойки (простая версия)

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

Это простая версия задачи. Единственное отличие в том, что в этой версии $n \leq 2000$. Вы можете делать взломы, только если решены обе версии задачи.

В линию выстроены n настоек, причем настойка 1 находится слева, а настойка n — справа. Каждая настойка увеличит ваше здоровье на a_i , если ее выпить. a_i может быть отрицательным, что означает, что настойка уменьшит ваше здоровье.

Вы начинаете с 0 здоровья и будете идти слева направо, от первой настойки до последней. Для каждой настойки вы можете выбрать, выпить ли ее. **Вы должны следить за тем, чтобы ваше здоровье всегда было неотрицательным.**

Какое наибольшее количество настоек вы можете выпить?

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 2000$) — количество настоек.

Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$), которые обозначают изменения здоровья после употребления данных настоек.

Выходные данные

Выведите одно целое число — максимальное количество настоек, которое вы можете выпить, чтобы ваше здоровье всегда было неотрицательным.

Пример

входные данные	Скопировать
6 4 -4 1 -3 1 -3	
выходные данные	Скопировать
5	

Примечание

В примере, вы можете выпить 5 настоек, приняв настойки 1, 3, 4, 5 и 6. Невозможно выпить все 6 настоек, потому что в какой-то момент ваше здоровье станет отрицательным.



В. Обнуление массива

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Вам дан массив a_1, a_2, \dots, a_n .

За одну операцию вы можете взять любые два элемента a_i и a_j ($i \neq j$) и уменьшить каждый из них на единицу.

Вам нужно проверить можно ли сделать все элементы массива равными нулю или нет.

Входные данные

Первая строка содержит одно целое число n ($2 \leq n \leq 10^5$) — размер массива.

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

Выходные данные

Выведите «YES», если можно сделать все элементы массива равными нулю. Иначе выведите «NO».

Примеры

входные данные	Скопировать
4 1 1 2 2	
выходные данные	Скопировать
YES	

входные данные	Скопировать
6 1 2 3 4 5 6	
выходные данные	Скопировать
NO	

Примечание

В первом примере можно сделать все элементы равными нулю за 3 операции:

- Уменьшить a_1 и a_2 ,
- Уменьшить a_3 и a_4 ,
- Уменьшить a_3 и a_4

Во втором примере можно показать, что сделать все элементы равными нулю нельзя.



В. Подпоследовательности Codeforces

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

Карл любит Codeforces и подпоследовательности. Он хочет составить строку из маленьких английских букв, которая содержит как минимум k подпоследовательностей codeforces. Из всех таких строк Карл хочет выбрать самую короткую.

Формально, подпоследовательность codeforces строки s — это подмножество из десяти символов s , которые образуют codeforces, если читать их в строке слева направо. Например, строка codeforces содержит одну подпоследовательность codeforces, а codeforcesisawesome содержит четыре подпоследовательности codeforces: **codeforces**isawesome, **code**for**ces**isawesome, **code**for**ces**isaw**esome**, **code**for**ces**isaw**esome**.

Помогите Карлу найти любую кратчайшую строку, которая содержит как минимум k подпоследовательностей codeforces.

Входные данные

В единственной строке записано одно целое число k ($1 \leq k \leq 10^{16}$).

Выходные данные

Выведите кратчайшую строку из маленьких английских букв, которая содержит не менее k подпоследовательностей codeforces. Если таких строк несколько, выведите любую из них.

Примеры

входные данные	Скопировать
1	
выходные данные	Скопировать
codeforces	

входные данные	Скопировать
3	
выходные данные	Скопировать
codeforcesss	



D. Уменьшение суммы цифр

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам задано целое положительное число n . За один ход вы можете увеличить n на единицу (то есть сделать $n := n + 1$). Ваша задача — найти минимальное количество ходов, которое надо совершить, чтобы сделать сумму цифр n не превышающей s .

Вам необходимо ответить на t независимых наборов тестовых данных.

Входные данные

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

Единственная строка набора тестовых данных содержит два целых числа n и s ($1 \leq n \leq 10^{18}$; $1 \leq s \leq 162$).

Выходные данные

На каждый набор тестовых данных выведите ответ: минимальное количество ходов, которое надо совершить, чтобы сделать сумму цифр n не превышающей s .

Пример

входные данные	Скопировать
5 2 1 1 1 500 4 217871987498122 10 100000000000000001 1	
выходные данные	Скопировать
8 0 500 2128012501878 899999999999999999	

D. Из бинарной строки в подпоследовательности

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

Вам задана бинарная строка s , состоящая из n нулей или единиц.

Ваша задача – разделить заданную строку на **минимальное** число **подпоследовательностей** таким образом, что *каждый* символ строки принадлежит ровно *одной* подпоследовательности и каждая подпоследовательность выглядит подобно «010101 . . .» или «101010 . . .» (т.е. подпоследовательность не должна содержать два соседних нуля или единицы).

Напомним, что подпоследовательность — это последовательность, которая может быть получена путем удаления из заданной последовательности с помощью удаления нуля или более элементов без изменения порядка остальных элементов. Например, подпоследовательностями «1011101» являются «0», «1», «11111», «0111», «101», «1001», но не «000», «101010» и «11100».

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 одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длину s . Вторая строка набора тестовых данных содержит n символов '0' и '1' — строку s .

Гарантируется, что сумма всех n не превосходит $2 \cdot 10^5$ ($\sum n \leq 2 \cdot 10^5$).

Выходные данные

Для каждого набора тестовых данных выведите ответ на него: первой строкой выведите одно целое число k ($1 \leq k \leq n$) — минимальное количество подпоследовательностей, на которые вы можете разделить строку s . Второй строкой выведите n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq k$), где a_i — номер подпоследовательности, к которой принадлежит i -й символ строки s .

Если существует несколько ответов, вы можете вывести любой.

Пример

входные данные	Скопировать
4 4 0011 6 111111 5 10101 8 01010000	
выходные данные	Скопировать
2 1 2 2 1 6 1 2 3 4 5 6 1 1 1 1 1 1 4 1 1 1 1 1 2 3 4	

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

С. К-полное слово

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

Слово s длины n называется k -полным, если

- s — палиндром, то есть $s_i = s_{n+1-i}$ для всех $1 \leq i \leq n$;
- s имеет период k , то есть $s_i = s_{k+i}$ для всех $1 \leq i \leq n - k$.

Например, «abaaba» — это 3-полное слово, а «abccba» нет.

Бобу вручили слово s длины n , состоящее только из строчных букв латинского алфавита, и целое число k такое, что n делится на k . Он хочет превратить слово s в любое k -полное слово.

Для этого Боб может выбирать некоторую позицию i ($1 \leq i \leq n$) и заменять букву на позиции i на любую другую строчную букву латинского алфавита.

Поэтому теперь Боба интересует минимальное количество позиций, буквы на которых ему необходимо заменить, чтобы превратить s в любое k -полное слово.

Обратите внимание, что Боб может сделать ноль изменений, если слово s уже k -полное.

Требуется ответить на t **независимых** наборов входных данных.

Входные данные

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

В первой строке каждого набора входных данных записаны два целых числа n и k ($1 \leq k < n \leq 2 \cdot 10^5$, n делится на k).

Во второй строке записано слово s длины n .

Гарантируется, что слово s состоит только из строчных букв латинского алфавита. Также гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число, обозначающее минимальное количество позиций, буквы на которых ему придется заменить, чтобы превратить s в любое k -полное слово.

Пример

входные данные	Скопировать
4 6 2 abaaba 6 3 abaaba 36 9 hippotomonstrosesquippedaliophobia 21 7 wudixiaoxingxingheclp	
выходные данные	Скопировать
2 0 23 16	

Примечание

В первом наборе входных данных одно из оптимальных решений — это «aaaaa».

Во втором наборе входных данных слово уже k -полное.

С. Восстановление бинарной строки

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

Рассмотрим следующий процесс. У вас есть бинарная строка (строка, состоящая только из символов 0 и 1) w длины n и число x . Вы создаете новую бинарную строку s длины n ; i -й символ новой строки s выбирается следующим образом:

- если символ w_{i-x} существует и равен 1, то s_i равно 1 (более формально, если $i > x$ и $w_{i-x} = 1$, то $s_i = 1$);
- если символ w_{i+x} существует и равен 1, то s_i равно 1 (более формально, если $i + x \leq n$ и $w_{i+x} = 1$, то $s_i = 1$);
- если оба описанных выше условия неверны, то s_i равно 0.

Вам заданы число x и строка s . Восстановите изначальную строку w .

Входные данные

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

Каждый набор входных данных содержит две строки. Первая строка содержит строку s ($2 \leq |s| \leq 10^5$, каждый символ строки s равен либо 0, либо 1). Вторая строка содержит целое число x ($1 \leq x \leq |s| - 1$).

Суммарная длина всех длин строк s во входных данных не превосходит 10^5 .

Выходные данные

На каждый набор входных данных выведите ответ:

- если не существует строки w , которая может породить строку s , то выведите -1 ;
- иначе, выведите бинарную строку w состоящую из $|s|$ символов. Если возможных ответов несколько — вы можете вывести любой из них.

Пример

входные данные	Скопировать
3 101110 2 01 1 110 1	
выходные данные	Скопировать
111011 10 -1	



С. Произведение 1 по модулю N

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

А вот первые слова маленького Эхаба: «Дано целое число n , найдите самую длинную подпоследовательность массива $[1, 2, \dots, n - 1]$, произведение элементов которой равно 1 по модулю n ». Решите эту задачу.

Последовательность b является подпоследовательностью массива a , если b может быть получена из a удалением нескольких (возможно, всех) элементов. Произведение элементов пустой подпоследовательности равно 1.

Входные данные

В единственной строке дано целое число n ($2 \leq n \leq 10^5$).

Выходные данные

В первой строке выведите длину найденной подпоследовательности.

Во второй строке выведите элементы подпоследовательности **в возрастающем порядке**.

Если существует несколько ответов, выведите любой.

Примеры

входные данные	Скопировать
5	
выходные данные	Скопировать
3 1 2 3	
входные данные	Скопировать
8	
выходные данные	Скопировать
4 1 3 5 7	

Примечание

В первом примере произведение элементов равно 6, что равно 1 по модулю 5. Единственная более длинная последовательность это $[1, 2, 3, 4]$. Произведение ее элементов равно 24, что равно 4 по модулю 5. Поэтому, ответ это $[1, 2, 3]$.

А. Дореми и её IQ

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

Дореми попросили протестировать n констестов. Констест i может быть протестирован только в день i . Сложность констеста i равна a_i . В начале показатель IQ Дореми равен q . В день i Дореми решает, протестирует ли она констест i или нет. Она может протестировать констест только в случае, если её текущий показатель IQ строго больше 0.

Если Дореми решает протестировать констест i в день i , то происходит следующее:

- если $a_i > q$, то Дореми чувствует себя недостаточно умной, поэтому q уменьшается на 1;
- иначе ничего не изменяется.

При решении не тестировать констест ничего не происходит.

Дореми хочет протестировать как можно больше констестов. Помогите Дореми найти оптимальное решение.

Входные данные

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

Первая строка каждого набора содержит целые числа n и q ($1 \leq n \leq 10^5$, $1 \leq q \leq 10^9$) — число констестов и показатель IQ Дореми в начале.

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

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 10^5 .

Выходные данные

Для каждого набора выведите бинарную строку s , где $s_i = 1$ если Дореми должна протестировать констест i , и $s_i = 0$ иначе. Число единиц в строке должно быть максимально возможным, и при этом она не должна тестировать констест, когда её показатель IQ меньше либо равен нулю.

Если существуют несколько решений, выведите любое из них.

Пример

входные данные	Скопировать
5 1 1 1 2 1 1 2 3 1 1 2 1 4 2 1 4 3 1 5 2 5 1 2 4 3	
выходные данные	Скопировать
1 11 110 1110 01111	

Примечание

В первом наборе входных данных Дореми тестирует единственный имеющийся констест. Её показатель IQ не уменьшается.

Во втором наборе входных данных Дореми тестирует оба имеющихся констеста. Её показатель IQ уменьшается на 1 после тестирования констеста 2.

В третьем наборе входных данных Дореми тестирует констесты 1 и 2. Её показатель IQ уменьшается до 0 после тестирования констеста 2, поэтому она не может протестировать констест 3.

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

С. Присвоить или уменьшить

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

Вам задан целочисленный массив a_1, a_2, \dots, a_n и целое число k .

За один шаг, вы можете

- либо выбрать некоторый индекс i и уменьшить a_i на единицу (сделать $a_i = a_i - 1$);
- либо выбрать два индекса i и j и присвоить элементу a_i значение a_j (сделать $a_i = a_j$).

За какое наименьшее количество шагов вы можете сделать сумму массива $\sum_{i=1}^n a_i \leq k$? (В массиве разрешены отрицательные элементы).

Входные данные

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

Во первой строке каждого набора заданы два целых числа n и k ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq k \leq 10^{15}$) — размер массива a и верхнее ограничение на сумму.

Во второй строке каждого набора заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — сам массив.

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

Выходные данные

Для каждого набора выведите одно число — наименьшее количество шагов для получения $\sum_{i=1}^n a_i \leq k$.

Пример

входные данные	Скопировать
4 1 10 20 2 69 6 9 7 8 1 2 1 3 1 2 1 10 1 1 2 3 1 2 6 1 6 8 10	
выходные данные	Скопировать
10 0 2 7	

Примечание

В первом наборе входных данных, вам нужно уменьшить a_1 10 раз, чтобы получить сумму не более $k = 10$.

Во втором наборе, сумма массива a уже не превосходит 69, а потому вам не нужно ничего менять.

В третьем наборе, вы можете, например:

- присвоить $a_4 = a_3 = 1$;
- уменьшить a_4 на единицу, и получить $a_4 = 0$.

В результате вы получите массив $[1, 2, 1, 0, 1, 2, 1]$ с суммой, не превосходящей 8, за $1 + 1 = 2$ шага.

В четвертом наборе, вы можете, например:

- выбрать a_7 и уменьшить его на один 3 раза; вы получите $a_7 = -2$;
- выбрать 4 элемента a_6, a_8, a_9 и a_{10} и присвоить им значение $a_7 = -2$.

В результате вы получите массив $[1, 2, 3, 1, 2, -2, -2, -2, -2, -2]$ с суммой, меньше или равной 1, за $3 + 4 = 7$ шагов.

E1. Удаляй и удлиняй (простая версия)

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

Это простая версия задачи. Единственное отличие — это ограничения на n и k . Вы можете делать взломы, только если все версии задачи решены.

У вас есть строка s , и вы можете выполнять над ней два типа операций:

- Удалить последний символ строки.
- Дублировать строку: $s := s + s$, где $+$ обозначает конкатенацию.

Вы можете использовать каждую операцию любое количество раз (возможно, ни одного).

Ваша задача — найти лексикографически наименьшую строку длины ровно k , которую можно получить, выполнив эти операции над строкой s .

Строка a лексикографически меньше строки b тогда и только тогда, когда выполняется одно из следующих условий:

- a является префиксом b , но $a \neq b$;
- В первой позиции, где a и b отличаются, строка a имеет букву, которая появляется раньше в алфавите, чем соответствующая буква в b .

Входные данные

Первая строка содержит два целых числа n, k ($1 \leq n, k \leq 5000$) — длину исходной строки s и длину желаемой строки.

Вторая строка содержит строку s , состоящую из n строчных английских букв.

Выходные данные

Выведите лексикографически наименьшую строку длины k , которая может быть получена путем выполнения операций над строкой s .

Примеры

входные данные	Скопировать
8 16 dbcadabc	
выходные данные	Скопировать
dbcadabcdbcadabc	

входные данные	Скопировать
4 5 abcd	
выходные данные	Скопировать
aaaaa	

Примечание

В первом тесте оптимально сделать одно дублирование: «dbcadabc» → «dbcadabcdbcadabc».

Во втором тесте оптимально удалить последние 3 символа, затем продублировать строку 3 раза, затем удалить последние 3 символов, чтобы строка имела длину k .

«abcd» → «abc» → «ab» → «a» → «aa» → «aaaa» → «aaaa» → «aaaa» → «aaaa» → «aaaa».

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

С. Нина и магическая матрица

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

У Нины есть матрица a размера $n \times n$, заполненная нулями. j -й элемент i -й строки матрицы a обозначается как $a_{i,j}$.

Она может выполнять операции двух типов с этой матрицей:

- Операция типа 1: выбрать целое число i от 1 до n и перестановку p_1, p_2, \dots, p_n целых чисел от 1 до n . Присвоить $a_{i,j} := p_j$ для всех $1 \leq j \leq n$ одновременно.
- Операция типа 2: выбрать целое число i от 1 до n и перестановку p_1, p_2, \dots, p_n целых чисел от 1 до n . Присвоить $a_{j,i} := p_j$ для всех $1 \leq j \leq n$ одновременно.

Нина хочет максимизировать сумму всех чисел в матрице $\sum_{i=1}^n \sum_{j=1}^n a_{i,j}$. Она просит вас найти способ выполнить операции так, чтобы эта сумма была максимальной. Поскольку она не хочет выполнять слишком много операций, вы должны предоставить решение с не более чем $2n$ операциями.

Перестановкой длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Входные данные

В первой строке дано одно целое число t ($1 \leq t \leq 500$) — количество наборов входных данных.

В единственной строке каждого набора входных данных дано одно целое число n ($1 \leq n \leq 500$) — размер матрицы a .

Гарантируется, что сумма n^2 по всем тестовым случаям не превосходит $5 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных на первой строке выведите два целых числа s и m ($0 \leq m \leq 2n$) — максимальную сумму чисел в матрице и количество операций в вашем решении.

В k -й из следующих m строк выведите описание k -й операции:

- целое число c ($c \in \{1, 2\}$) — тип k -й операции;
- целое число i ($1 \leq i \leq n$) — строка или столбец, к которому применяется k -я операция;
- перестановка p_1, p_2, \dots, p_n целых чисел от 1 до n — перестановка, используемая в k -й операции.

Обратите внимание, что вам не нужно минимизировать количество использованных операций, вам просто нужно использовать не более чем $2n$ операций. Можно показать, что максимальная возможная сумма всегда может быть получена не более чем за $2n$ операций.

Пример

входные данные	Скопировать
2 1 2	
выходные данные	Скопировать
1 1 1 1 1 7 3 1 1 1 2 1 2 1 2 2 1 1 2	

Примечание

В первом наборе входных данных максимальная сумма $s = 1$ может быть получена за 1 операцию присвоением $a_{1,1} := 1$.

Во втором наборе входных данных максимальная сумма $s = 7$ может быть получена за 3 операции следующим образом:



0	0
---	---

0	0
---	---

1	2
---	---

2	2
---	---

Можно показать, что невозможно сделать сумму чисел в матрице большей 7.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
 Соревнования по программированию 2.0
 Время на сервере: 30.06.2025 10:11:19^{UTC+5} (j1).
 Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

А. Весело считать Кенгуру

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

Даны n кенгуру с сумками. У каждого кенгуру есть размер (целое число). Кенгуру может поместиться в сумке другого кенгуру тогда и только тогда, когда размер кенгуру-носителя как минимум в два раза больше размера кенгуру-пассажира.

Каждый кенгуру может нести не более одного кенгуру, а кенгуру-пассажир не может носить никаких кенгуру.

Кенгуру-пассажира не видно, когда он в сумке кенгуру-носителя. Пожалуйста, разработайте такой план рассадки кенгуру, чтобы было видно как можно меньше кенгуру.

Входные данные

В первой строке записано единственное целое число — n ($1 \leq n \leq 5 \cdot 10^5$). Каждая из следующих n строк содержит целое число s_i — размер i -го кенгуру ($1 \leq s_i \leq 10^5$).

Выходные данные

Выведите единственное целое число — оптимальное количество видимых кенгуру.

Примеры

входные данные	Скопировать
8 2 5 7 6 9 8 4 2	
выходные данные	Скопировать
5	

входные данные	Скопировать
8 9 1 6 2 6 5 8 3	
выходные данные	Скопировать
5	



А. Сбалансируйте биты

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

Скобочная последовательность называется сбалансированной, если ее можно превратить в корректное математическое выражение путем добавления символов '+' и '1'. Например, последовательности « $((()))$ », « $()$ » и « $((())())$ » являются сбалансированными, а « $)()$ », « $((())$ » и « $((()))()$ » — нет.

Вам дана бинарная строка s длины n . Постройте две сбалансированные скобочные последовательности a и b длины n такие, что для всех $1 \leq i \leq n$ выполняется:

- если $s_i = 1$, то $a_i = b_i$
- если $s_i = 0$, то $a_i \neq b_i$

Если это невозможно, вы должны это определить.

Входные данные

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

В первой строке каждого набора входных данных содержится одно целое число n ($2 \leq n \leq 2 \cdot 10^5$, n — четное).

Следующая строка содержит строку s длины n , состоящую из символов 0 и 1.

Сумма n во всех тестовых случаях не превышает $2 \cdot 10^5$.

Выходные данные

Если такие две сбалансированные скобочные последовательности существуют, выведите «YES» в первой строке, иначе выведите «NO». Вы можете выводить каждый символ в любом регистре.

Если ответ «YES», в двух следующих строках выведите сбалансированные скобочные последовательности a и b , удовлетворяющие условиям.

Если есть несколько решений, то можно вывести любое.

Пример

входные данные	Скопировать
3 6 101101 10 1001101101 4 1100	
выходные данные	Скопировать
YES () () YES () () YES () () NO	

Примечание

В первом наборе входных данных $a = «()()()»$ и $b = «((()))»$. Символы равны в позициях 1, 3, 4 и 6, ровно в тех позициях, где $s_i = 1$.

Во втором наборе входных данных $a = «()()((()))»$ и $b = «((())()())»$. Символы равны в позициях 1, 4, 5, 7, 8, 10, ровно в тех позициях, где $s_i = 1$.

В третьем наборе входных данных решения нет.

С. Ехаб и префиксные МЕХы

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

Для данного массива a длины n найдите другой массив b длины n такой, что:

- для каждого i ($1 \leq i \leq n$) $MEX(\{b_1, b_2, \dots, b_i\}) = a_i$.

MEX множества целых чисел равен наименьшему неотрицательному целому числу, которое не принадлежит этому множеству.

Если такого массива не существует, определите это.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — длина массива a .

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq i$) — элементы массива a . **Гарантируется, что $a_i \leq a_{i+1}$ для $1 \leq i < n$.**

Выходные данные

Если такого массива нет, выведите одну строку, содержащую -1 .

В противном случае выведите одну строку, содержащую n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^6$).

Если есть несколько ответов, выведите любой.

Примеры

входные данные	Скопировать
3 1 2 3	
выходные данные	Скопировать
0 1 2	
входные данные	Скопировать
4 0 0 0 2	
выходные данные	Скопировать
1 3 4 0	
входные данные	Скопировать
3 1 1 3	
выходные данные	Скопировать
0 2 1	

Примечание

Во втором примере допустимы другие ответы, например, $[1, 1, 1, 0]$.



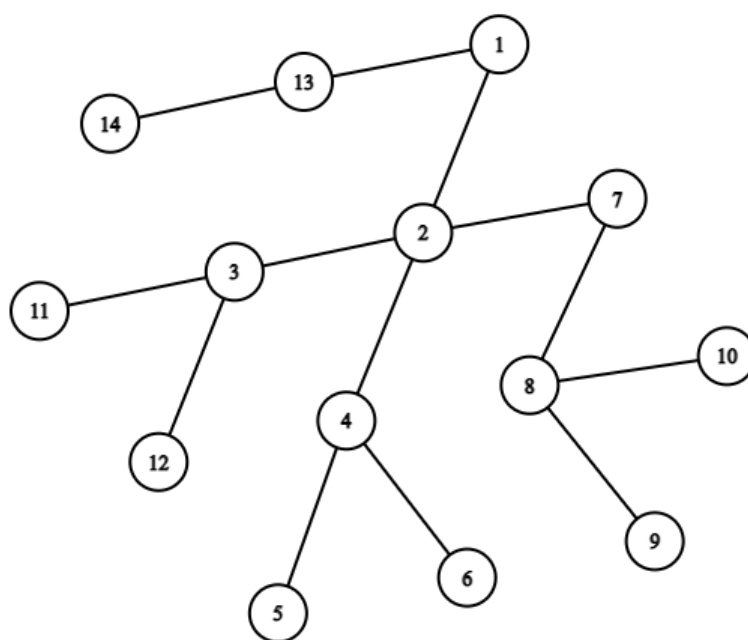
Е. Садовник и дерево

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 256 мегабайт

Дерево — это неориентированный связный граф, в котором отсутствуют циклы. В этой задаче речь идет о некорневых деревьях.

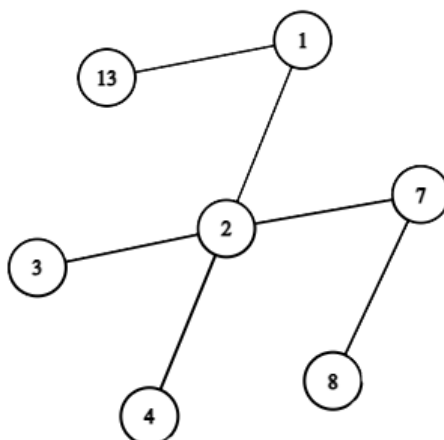
Лист дерева — это вершина, которая соединена **не более чем с одной** другой вершиной.

Садовник Виталий вырастил дерево из n вершин. Он решил подстричь дерево. Для этого он совершает несколько операций. За одну операцию он удаляет **все** листья дерева.



Пример дерева.

Например, рассмотрим дерево, изображённое на рисунке выше. На рисунке ниже приведён результат применения к дереву ровно одной операции.



Результат применения операции «удалить все листья».

Обратите внимание на особенные случаи применения операций:

- применение операции к пустому дереву (из 0 вершин) не меняет его;
- применение операции к дереву из одной вершины приводит к удалению этой вершины (т. е. одна эта вершина считается листом);

- применение операции к дереву из двух вершин приводит к удалению обеих вершин (обе вершины считаются листьями).

Виталий применил к дереву последовательно k операций. Сколько вершин в нём осталось?

Входные данные

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

Перед каждым набором входных данных расположена пустая строка.

Каждый набор данных состоит из нескольких строк. Первая строка набора данных содержит два целых числа n и k ($1 \leq n \leq 4 \cdot 10^5$, $1 \leq k \leq 2 \cdot 10^5$) — количество вершин в дереве и количество операций. Далее идут $n - 1$ строк, каждая содержит два целых числа u и v ($1 \leq u, v \leq n$, $u \neq v$) — две вершины, которые соединены ребром. Гарантируется, что заданный граф является деревом, в нём отсутствуют петли и кратные рёбра. Гарантируется, что сумма n из всех наборов входных данных не превосходит $4 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите в отдельной строке одно целое число — количество вершин, которое осталось в дереве после применения k операций.

Пример

входные данные	Скопировать
<pre>6 14 1 1 2 2 3 2 4 4 5 4 6 2 7 7 8 8 9 8 10 3 11 3 12 1 13 13 14 2 200000 1 2 3 2 1 2 2 3 5 1 5 1 3 2 2 1 5 4 6 2 5 1 2 5 5 6 4 2 3 4 7 1 4 3 5 1 1 3 6 1 1 7 2 1</pre>	
выходные данные	Скопировать
<pre>7 0 0 3 1 2</pre>	

Примечание

Первый набор входных данных разобран в условии.

Во втором наборе задано дерево из двух вершин. К нему применяются 200000 операций. Первая удаляет обе вершины, остальные операции не изменяют дерево.

В третьем наборе входных данных дано дерево из трёх вершин. В результате применения первой операции в нём остаётся всего 1 вершина (с номером 2), в результате второй операции дерево становится пустым.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 30.06.2025 10:11:29^{UTC+5} (j1).
Мобильная версия, переключиться на [desktopную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ІТМО

С. Получи строку

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

Вам заданы две строки s и t , состоящие из строчных букв латинского алфавита. Также у вас есть строка z , которая изначально пуста. Вы хотите, чтобы строка z стала равна строке t . Для этого вы можете выполнять следующую операцию: добавить любую подпоследовательность строки s в конец строки z . Подпоследовательность — это последовательность, которая получается из данной путем удаления нуля или более ее элементов. Например, если $z = ac$, $s = abcde$, вы можете превратить z в следующие строки за одну операцию:

1. $z = acace$ (если выберете подпоследовательность ace);
2. $z = acbcd$ (если выберете подпоследовательность bcd);
3. $z = acbce$ (если выберете подпоследовательность bce).

Обратите внимание, что строка s не меняется после операций.

Посчитайте минимальное количество операций, необходимое для того, чтобы превратить строку z в строку t .

Входные данные

Первая строка содержит единственное число T ($1 \leq T \leq 100$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит строку s ($1 \leq |s| \leq 10^5$), состоящую из строчных букв латинского алфавита.

Вторая строка каждого набора входных данных содержит строку t ($1 \leq |t| \leq 10^5$), состоящую из строчных букв латинского алфавита.

Гарантируется, что суммарная длина строк s и t во входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

На каждый набор входных данных выведите одно число — минимальное количество операций, необходимое для того, чтобы превратить строку z в строку t . Если это невозможно выведите -1 .

Пример

входные данные	Скопировать
3 aabce ace abacaba aax ty yut	
выходные данные	Скопировать
1 -1 3	



С. Минимизируйте число

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

Вам задано длинное число a , состоящее из n цифр (n от 1 до $3 \cdot 10^5$ включительно). Оно может содержать лидирующие нули.

Вы можете поменять местами две соседние цифры, если они имеют разную четность (т.е. они имеют разный остаток от деления на 2).

Например, если $a = 032867235$ вы можете получить следующие числа за одну операцию:

- 302867235 если поменяете местами первую и вторую цифры;
- 023867235 если поменяете местами вторую и третью цифры;
- 032876235 если поменяете местами пятую и шестую цифры
- 032862735 если поменяете местами шестую и седьмую цифры;
- 032867325 если поменяете местами седьмую и восьмую цифры.

Обратите внимание, что вы не можете поменять местами цифры на позициях 2 и 4, потому что эти позиции не соседние. Так же, вы не можете поменять местами цифры на позициях 3 и 4, потому что эти цифры имеют одинаковую четность.

Вы можете выполнять любое (возможно, нулевое) количество таких операций.

Найдите минимальное число, которое вы можете получить.

Обратите внимание, что ответ так же может содержать лидирующие нули.

Входные данные

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

Единственная строка каждого набора содержит длинное число a , его длина n в отрезке от 1 до $3 \cdot 10^5$ включительно.

Гарантируется, что сумма всех n не превосходит $3 \cdot 10^5$.

Выходные данные

На каждый набор входных данных выведите ответ — минимальное число, которое вы можете получить.

Пример

входные данные	Скопировать
3 0709 1337 246432	
выходные данные	Скопировать
0079 1337 234642	

Примечание

В первом наборе вы можете выполнить следующую последовательность операций (пара цифр, которую вы меняете местами, выделена): 0709 → 0079.

Во втором наборе изначальное число является оптимальным.

В третьем наборе вы можете выполнить следующую последовательность операций:

246432 → 246342 → 243642 → 234642.

С. Наименьшая префиксная сумма

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

У Балтика, известного шахматиста, а также математика, есть массив a_1, a_2, \dots, a_n , и он хочет выполнить следующую операцию несколько (возможно, 0) раз:

- Выбрать индекс i ($1 \leq i \leq n$);
- умножить a_i на -1 , то есть присвоить $a_i := -a_i$.

Любимое число Балтика равно m , поэтому он хочет, чтобы значение $a_1 + a_2 + \dots + a_m$ было наименьшим среди всех непустых префиксных сумм. Формально, для всех $k = 1, 2, \dots, n$ должно выполняться

$$a_1 + a_2 + \dots + a_k \geq a_1 + a_2 + \dots + a_m.$$

Обратите внимание, что может существовать несколько наименьших префиксных сумм, и необходимо только, чтобы $a_1 + a_2 + \dots + a_m$ была одной из них.

Помогите Балтику найти минимальное количество операций, необходимое, чтобы сумма $a_1 + a_2 + \dots + a_m$ стала наименьшей префиксной суммой. Можно показать, что необходимая последовательность операций всегда существует.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и m ($1 \leq m \leq n \leq 2 \cdot 10^5$) — размер массива и его любимое число.

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

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

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимально необходимое число операций.

Пример

входные данные	Скопировать
6 4 3 -1 -2 -3 -4 4 3 1 2 3 4 1 1 1 5 5 -2 3 -5 1 -20 5 2 -2 3 -5 -5 -20 10 4 345875723 -48 384678321 -375635768 -35867853 -35863586 -358683842 -81725678 38576 -357865873	
выходные данные	Скопировать
1 1 0 0 3 4	

Примечание

В первом примере можно выполнить операцию $a_4 := -a_4$. Массив становится равным $[-1, -2, -3, 4]$, а префиксные суммы $[a_1, a_1 + a_2, a_1 + a_2 + a_3, a_1 + a_2 + a_3 + a_4]$ становятся равными $[-1, -3, -6, -2]$. Поэтому $a_1 + a_2 + a_3 = -6$ является наименьшей из всех префиксных сумм.

Во втором примере можно выполнить операцию $a_3 := -a_3$. Массив становится равным $[1, 2, -3, 4]$, префиксные суммы равны $[1, 3, 0, 4]$.

В третьем и четвертом примерах $a_1 + a_2 + \dots + a_m$ уже является наименьшей префиксной суммой, нет необходимости

выполнять операции.

В пятом примере одна из подходящих последовательностей операций следующая:

- $a_3 := -a_3$,
- $a_2 := -a_2$,
- $a_5 := -a_5$.

Массив становится равным $[-2, -3, 5, -5, 20]$, его префиксные суммы равны $[-2, -5, 0, -5, 15]$. Обратите внимание, что и $a_1 + a_2 = -5$, и $a_1 + a_2 + a_3 + a_4 = -5$ — минимальные префиксные суммы (и это корректное решение).

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 30.06.2025 10:11:34^{UTC+5} (n2).
Мобильная версия, переключиться на [деSKTOPННУЮ](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

С. Кеши устраивает вечеринку

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

Кеши устраивает вечеринку и хочет, чтобы все на ней были счастливы.

У него есть n друзей. У его i -го друга есть i долларов.

Если пригласить i -го друга на вечеринку, то он будет счастлив только в том случае, если не более a_i человек на вечеринке будут строго богаче его и не более b_i человек будут строго беднее его.

Кеши хочет пригласить как можно больше людей. Найдите максимальное число людей, которых он может пригласить на вечеринку, чтобы каждый приглашенный был счастлив.

Входные данные

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

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество друзей Кеши.

i -я из следующих n строк содержит два целых числа a_i и b_i ($0 \leq a_i, b_i < n$).

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

Выходные данные

Для каждого набора входных данных выведите максимальное количество людей, которых может пригласить Кеши.

Пример

входные данные	Скопировать
<pre>3 3 1 2 2 1 1 1 2 0 0 0 1 2 1 0 0 1</pre>	
выходные данные	Скопировать
<pre>2 1 2</pre>	

Примечание

В первом наборе входных данных он приглашает первого и второго человека. Если он пригласит всех, то третий человек не будет счастлив, потому что будет более чем 1 человек беднее его.



D. Таблица-00100

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

Безумный ученый Dr.Jubal придумал задачу по программированию. Попробуйте решить ее!

Вам даны целые числа n, k . Постройте таблицу A , имеющую размер $n \times n$ и состоящую из целых чисел 0 или 1. Должно быть выполнено очень важное условие: сумма всех элементов в таблице равна k . Другими словами, количество чисел 1 в таблице равно k .

Давайте определим:

- $A_{i,j}$ как число, стоящее в i -й строке и j -м столбце.
- $R_i = A_{i,1} + A_{i,2} + \dots + A_{i,n}$ (для всех $1 \leq i \leq n$).
- $C_j = A_{1,j} + A_{2,j} + \dots + A_{n,j}$ (для всех $1 \leq j \leq n$).
- Другими словами, R_i это суммы чисел в строках и C_j это суммы чисел в столбцах таблицы A .
- Для таблицы A определим значение $f(A) = (\max(R) - \min(R))^2 + (\max(C) - \min(C))^2$ (здесь для последовательности целых чисел X мы определяем $\max(X)$ как максимальное число в X и $\min(X)$ как минимальное число в X).

Найдите любую таблицу A , удовлетворяющую описанному условию. Среди всех таких таблиц найдите такую, для которой значение $f(A)$ минимально возможное. Среди всех таких таблиц найдите любую.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Следующие t строк содержат описания наборов входных данных.

Для каждого набора входных данных в единственной строке находится два целых числа n, k ($1 \leq n \leq 300, 0 \leq k \leq n^2$).

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

Выходные данные

Для каждого набора входных данных сначала выведите минимальное возможное значение $f(A)$ среди всех таблиц, для которых условие выполнено.

После этого, выведите n строк, каждая содержит по n символов. j -й символ в i -й строке должен быть равен $A_{i,j}$.

Если есть несколько возможных решений, вы можете вывести любое.

Пример

входные данные	Скопировать
4 2 2 3 8 1 0 4 16	
выходные данные	Скопировать
0 10 01 2 111 111 101 0 0 0 1111 1111 1111 1111	

Примечание

В первом наборе входных данных сумма всех чисел таблицы равна 2, поэтому условие выполнено. $R_1 = 1, R_2 = 1$ и $C_1 = 1, C_2 = 1$. Тогда $f(A) = (1 - 1)^2 + (1 - 1)^2 = 0$, что является минимальным возможным значением $f(A)$.

С. Сбалансированные кучки камней

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

В ряд находятся n кучек камней. В i -й кучке изначально h_i камней. Вы хотите изменить число камней в кучках, выполнив следующий процесс один раз:

- Вы идете по кучкам от 3-й до n -й, в таком порядке.
- Пусть i — номер текущей кучки.
- Вы можете выбрать целое число d ($0 \leq 3 \cdot d \leq h_i$), переместить d камней из i -й кучки в $(i - 1)$ -ю кучку, и $2 \cdot d$ камней из i -й кучки в $(i - 2)$ -ю.
- Таким образом, h_i уменьшается на $3 \cdot d$, h_{i-1} увеличивается на d , и h_{i-2} увеличивается на $2 \cdot d$.
- Вы можете выбирать различные или одинаковые d для различных операций. Некоторые кучки могут стать пустыми, но они все еще считаются за кучки.

Какое наибольшее число камней в наименьшей кучке может получиться после завершения процесса?

Входные данные

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 2 \cdot 10^5$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка каждого набора входных данных содержит целое число n ($3 \leq n \leq 2 \cdot 10^5$).

Вторая строка каждого набора содержит n целых чисел $h_1, h_2, h_3, \dots, h_n$ ($1 \leq h_i \leq 10^9$).

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

Выходные данные

Для каждого набора входных данных выведите максимальное число камней, которое может содержать наименьшая по количеству камней кучка после завершения процесса.

Пример

входные данные	Скопировать
4	
4	
1 2 10 100	
4	
100 100 100 1	
5	
5 1 1 1 8	
6	
1 2 3 4 5 6	
выходные данные	Скопировать
7	
1	
1	
3	

Примечание

В первом примере изначально размеры кучек равны $[1, 2, 10, 100]$. Мы можем перемещать камни следующим образом.

- переместить 3 и 6 камней с 3-й кучки на 2-ю и 1-ю кучки соответственно. Размеры кучек будут равны $[7, 5, 1, 100]$;
- Переместить 6 и 12 камней с последней кучки на 3-ю и 2-ю кучки соответственно. Размеры кучек будут равны $[7, 17, 7, 82]$.

Во втором наборе размер последней кучки равен 1 и мы не можем его увеличить.

В третьем наборе оптимально не перемещать никакие камни.

В четвертом примере можно достичь состояния, в котором размеры кучек равны $[3, 5, 3, 4, 3, 3]$.

D. Выборы

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

В Берляндии проходят выборы. На выборах участвуют n кандидатов, пронумерованных от 1 до n . У i -го из них есть a_i фанатов, которые проголосуют за него. Также дополнительно есть c людей, которые не определились с любимым кандидатом, назовем их *неопределившимися*. Неопределившиеся люди проголосуют за кандидата с наименьшим номером.

На выборах побеждает кандидат, который набрал максимальное количество голосов, а если максимальное количество голосов набрали несколько кандидатов, то среди таких побеждает кандидат с наименьшим номером.

Такие выборы вам показались слишком скучными и предсказуемыми, поэтому вы решили не допустить некоторых кандидатов к ним. Если вы не допустите кандидата с номером i к выборам, то все a_i его фанатов станут неопределившимися, и на выборах проголосуют за допущенного кандидата с наименьшим номером.

Вам стало интересно узнать для каждого i от 1 до n , какое минимальное количество кандидатов надо не допустить к выборам, чтобы кандидат с номером i выиграл выборы.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leq t \leq 2 \cdot 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и c ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq c \leq 10^9$) — количество кандидатов на выборах и количество неопределившихся людей.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — количество фанатов у каждого кандидата.

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

Выходные данные

Для каждого набора входных данных выведите n целых чисел, i -е из которых должно быть равно минимальному количеству кандидатов, которых нужно не допустить к выборам, чтобы кандидат с номером i победил.

Пример

входные данные	Скопировать
5 3 1 2 0 3 2 3 0 10 5 3 5 4 3 2 1 4 5 3 10 7 1 6 0 2 2 2 3 3 3	
выходные данные	Скопировать
0 1 2 1 0 0 1 2 3 4 1 0 2 3 1 1 2 0 4 5	

Примечание

В первом наборе входных данных:

- Если допустить всех кандидатов, то кандидат с номером 1 наберёт 3 голоса (за него проголосует 1 неопределившийся человек), кандидат с номером 2 наберёт 0 голосов, кандидат с номером 3 наберёт 3 голоса. Таким образом, выиграет кандидат с номером 1 (он набрал столько же голосов, сколько кандидат 3, но его номер меньше), поэтому для него ответ равен 0.
- Если не допустить кандидата с номером 1, то его 2 фаната станут неопределившимися. Тогда кандидат с номером 2 наберёт 3 голоса (за него проголосуют 3 неопределившихся человека) и кандидат с номером 3 наберёт 3 голоса. Таким образом, выиграет кандидат с номером 2 (он набрал столько же голосов, сколько кандидат 3, но его номер меньше), поэтому для него ответ равен 1.