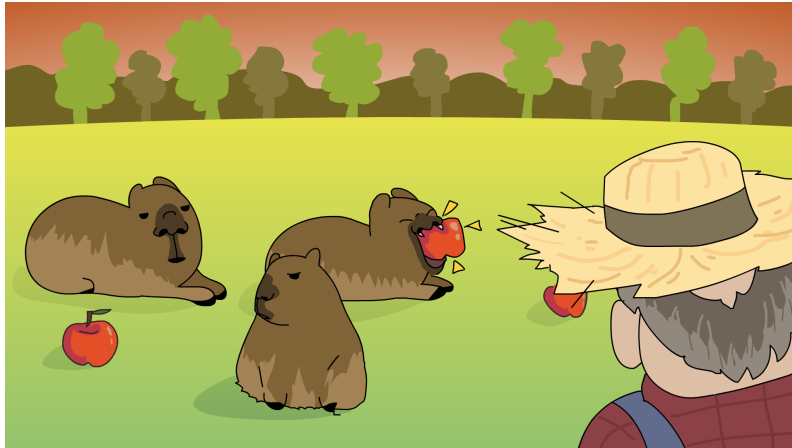


Задача А. Утро в марсианском саду

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Каждое утро лучший садовник Марса начинает одинаково. Он просыпается, завтракает согласно своей яблочной диете и выгуливает марсианских капибар. Далее по плану он должен сделать менее важные дела, а именно, разобраться с бумагами. В частности, ему необходимо записать, сколько яблок какого сорта у него растёт.



Все марсианские сорта яблок называются очень своеобразно, поэтому Казимир Казимирович, как хороший садовник, записал названия трех новых высаженных сортов в текстовый файл. Но из-за небольшой ошибки все названия склеились в одну строку s !

Обозначим названия сортов яблок как непустые строки s_1 , s_2 и s_3 . Тогда текстовый файл содержит строки s_1 , s_2 и s_3 подряд без пробелов между ними. Например, если сорта имеют названия «petr», «anna» и «vasiliy», то записанная в файле строка s будет выглядеть как «petrannavasiliy».

Однако Казимир запомнил один факт про эти сорта: либо строка s_2 лексикографически не больше строк s_1 и s_3 одновременно, либо строка s_2 лексикографически не меньше строк s_1 и s_3 одновременно. Иными словами, либо выполнено $s_1 \leq s_2$ и $s_3 \leq s_2$, либо выполнено $s_2 \leq s_1$ и $s_2 \leq s_3$ (а возможно, и оба условия одновременно). Здесь \leq обозначает лексикографическое «меньше или равно» для строк. Таким образом, $x \leq y$ значит, что строки должны быть либо равны, либо строка x должна стоять раньше в словаре, чем строка y . Более подробное объяснение этой операции см. в разделе «Замечание».

Рассмотрим некоторые примеры. Вместо s_1 , s_2 и s_3 можно подставить «petr», «vasiliy» и «anna», поскольку тогда выполняется первая часть условия («petr» \leq «vasiliy» и «anna» \leq «vasiliy»). Также корректны и «petr», «anna» и «vasiliy», поскольку выполнена вторая часть условия (т. е. «anna» \leq «petr» и «anna» \leq «vasiliy»). В то же время тройка «anna», «petr» и «vasiliy» не подходит под описанное выше свойство.

Помогите Казимиру найти любую тройку названий сортов, которая удовлетворяет описанному выше свойству. При этом запись этих названий без пробелов должна давать заданную строку s .

Формат входных данных

В единственной строке входных данных находится строка s ($3 \leq |s| \leq 3 \cdot 10^5$) — названия сортов, записанные без пробелов.

Формат выходных данных

Выведите три строки s_1 , s_2 и s_3 , разделенные пробелами — названия сортов, при записи которых без пробелов получается строка s . При этом должно выполняться либо $s_1 \leq s_2$ и $s_3 \leq s_2$, либо $s_2 \leq s_1$ и $s_2 \leq s_3$.

Если способов восстановить названия несколько, то выведите любой из них. Если названия восстановить невозможно, то выведите «:(» (без кавычек).

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$ s = 3$	8	
2	$ s \leq 100$	17	1
3	$ s \leq 300$	19	1 - 2
4	$ s \leq 3000$	25	1 - 3
5	Нет дополнительных ограничений	31	1 - 4

Примеры

стандартный ввод	стандартный вывод
petrvasiliyanna	petr vasilij anna
petrannavasiliy	petr anna vasilij
abc	:(
aaa	a a a

Замечание

Рассмотрим подробнее алгоритм лексикографического сравнения.

- Если строки полностью совпадают, то они равны
- Если одна строка является префиксом другой, то эта строка лексикографически меньше
- Иначе пусть первые k символов этих строк равны ($k \geq 0$), а символы на $(k + 1)$ -й позиции различаются. Тогда лексикографически меньше та строка, у которой $(k + 1)$ -й символ стоит раньше в алфавитном порядке.

Тогда получаем, что, например, «abc» ≤ «abc», «a» ≤ «aa», «abx» ≤ «baa», «aabx» ≤ «aafd».

Теперь перейдем к примерам.

Первые два примера разобраны выше.

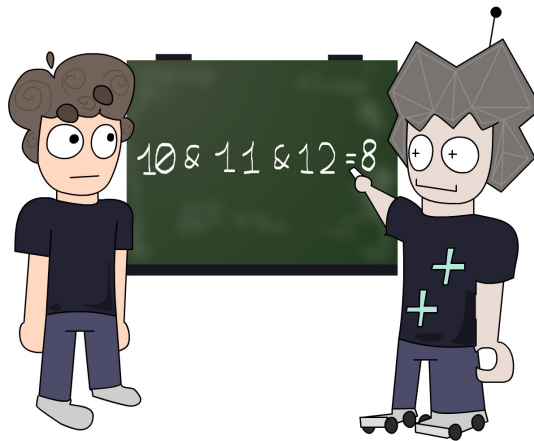
В третьем примере единственный возможный способ разбить строку s на три строки — это «a», «b», «c». Но такое разбиение не удовлетворяет описанному в условии свойству, поэтому требуется вывести «:(».

В четвертом примере можно заметить, что разбиение удовлетворяет двум условиям сразу (т. е. $s_1 \leq s_2$, $s_3 \leq s_2$, $s_2 \leq s_1$ и $s_2 \leq s_3$ верны одновременно).

Задача В. Битва умов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Робот Petya++ и его друг, Петя, столкнулись в интеллектуальном поединке, решая задачи друг друга. Petya++ уже с легкостью решил задачу Пети, и теперь настало время Пети решить задачу от робота.



Petya++ придумал два целых числа, n и x , а также записал следующую формулу:

$$n \& (n + 1) \& (n + 2) \& \dots \& m = x,$$

Задача Пети — найти минимальное m ($m \geq n$) такое, что равенство на доске выполняется. В формуле выше $\&$ обозначает операцию побитового И (см. также раздел «Замечание»).

Естественно, Петя с этой задачей не справился и решил попросить вашей помощи. А справитесь ли вы с задачей Petya++?

Формат входных данных

В единственной строке входных данных записаны два целых числа n и x ($0 \leq n, x \leq 2^{60}$) — числа, придуманные Petya++.

Формат выходных данных

Выведите одно целое число m ($n \leq m \leq 2^{60}$) — минимальное число такое, что равенство на доске будет выполняться.

Если равенство не выполнится ни при каком m , то выведите вместо этого -1 .

Гарантируется, что если требуемое m существует, то оно не превосходит 2^{60} .

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n, x \leq 2^{10}, n = x$	6	
2	$n, x \leq 2^{10}$	11	1
3	$n, x \leq 2^{20}$	21	1 - 2
4	$n, x \leq 2^{30}$	17	1 - 3
5	$x = 0$	9	
6	x — степень двойки	13	
7	Нет дополнительных ограничений	23	1 - 6

Примеры

стандартный ввод	стандартный вывод
10 8	12
10 10	10
10 42	-1
20 16	24

Замечание

И — операция, которая на вход принимает два бита и возвращает бит 1, если оба бита на входе равны 1, и 0 в противном случае. Например: $1 \& 0 = 0$, $1 \& 1 = 1$. Для применения побитового И двух чисел эти числа сначала переводят в двоичную систему счисления, а затем применяют И к каждому из разрядов. Например, $6 \& 3 = 2$, поскольку $6 = 110_2$, $3 = 11_2$. Применяв И поразрядно, получаем $2 = 10_2$:

$$\begin{array}{r} 110_2 \\ \& 11_2 \\ \hline 10_2 \end{array}$$

В языке программирования Pascal побитовое И чисел a и b обозначается $a \text{ and } b$, а в языках Python и C++ — $a \& b$.

Теперь рассмотрим примеры.

В первом примере $10 \& 11 = 10$, но $10 \& 11 \& 12 = 8$, поэтому ответ равен 12.

Во втором примере $10 = 10$, поэтому ответ равен 10.

В третьем примере можно убедиться, что требуемого m не существует, поэтому необходимо вывести -1 .

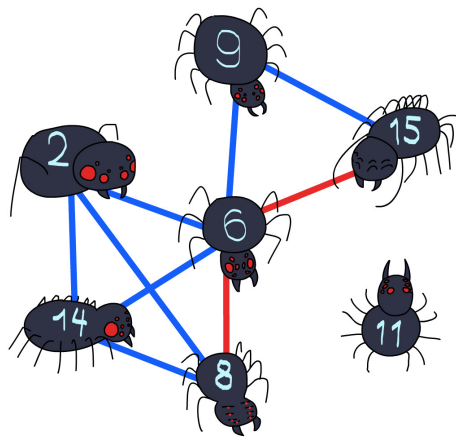
Задача С. Паутина сплетней

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Марс всегда славился своим богатым животным миром. Например, там водится особый вид пауков — Двоичные пауки. Их поведение окутано тайной, однако каждый марсианин знает — двоичные пауки всегда рады поговорить друг с другом о чем-то общем, в частности, об общем делителе.

Рассмотрим общение пауков на примере некоторой паутины. Пусть на ней сидит n пауков, i -й из которых имеет a_i лапок.

Известно, что i -й и j -й пауки будут общаться только в случае, если у их количества лапок найдется общий делитель, т. е. существует некоторое число $k \geq 2$ такое, что a_i и a_j одновременно делятся на k без остатка.



Двоичные пауки очень любят сплетни. Каждый паук может рассказать сплетню любому другому пауку, с которым он общается. Тот, в свою очередь также может передать ее любому пауку, с которым он общается и так далее, пока сплетня не дойдет до паука, с кем изначальный паук хотел ей поделиться.

Рассмотрим пример.

Пусть паук с восемью лапками хочет рассказать сплетню пауку с 15-ю лапками. Напрямую он этого сделать не может, потому что $\text{НОД}(8, 15) = 1$. Зато он может передать ее через паука с шестью лапками, поскольку $\text{НОД}(8, 6) = 2$ и $\text{НОД}(6, 15) = 3$. Таким образом, сплетня дойдет за две секунды.

Прямо сейчас s -й паук хочет рассказать сплетню t -му пауку. Двоичные пауки ценят свое время, поэтому сплетни всегда доходят самым оптимальным, т. е. кратчайшим, путем. От вас требуется сообщить, дойдет ли сплетня до адресата и если дойдет, то каким маршрутом.

Формат входных данных

В первой строке входных данных находится целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество пауков на паутине.

Во второй строке входных данных находится n целых чисел a_i ($1 \leq a_i \leq 3 \cdot 10^5$) — количество лапок у i -го паука.

В третьей строке входных данных находится два целых числа s и t ($1 \leq s, t \leq n$) — пауки, между которыми необходимо передать сплетню.

Формат выходных данных

Если передать сплетню между заданной парой пауков невозможно, выведите -1 .

Иначе в первой строке выходных данных выведите целое число t ($t \geq 1$) — количество пауков, которые участвуют в передаче сплетни (т. е. минимальное время доставки сплетни в секундах плюс

один). Во второй строке выведите t различных целых чисел b_i ($1 \leq b_i \leq n$) — номера пауков, через которых должна следовать сплетня, в порядке ее следования от отправителя к получателю.

Если вариантов построения оптимального маршрута для сплетни несколько, то выведите любой из них.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n = 2$	4	
2	$n \leq 2000$	21	1
3	$a_i \leq 2000$	10	
4	$a_i \leq 10^4$	12	3
5	$a_i = i$	27	
6	Нет дополнительных ограничений	26	1 - 5

Примеры

стандартный ввод	стандартный вывод
7 2 14 9 6 8 15 11 5 6	3 5 4 6
7 2 14 9 6 8 15 11 5 7	-1

Замечание

Первый пример разобран выше. В нем видно, что сплетню от паука номер 5 (с восемью лапками) к пауку номер 6 (с 15-ю лапками) оптимально передать через паука номер 4 (с шестью лапками).

Во втором примере паук номер 7 (с 11 лапками) ни с кем не дружит, поэтому ему невозможно передать сплетню.

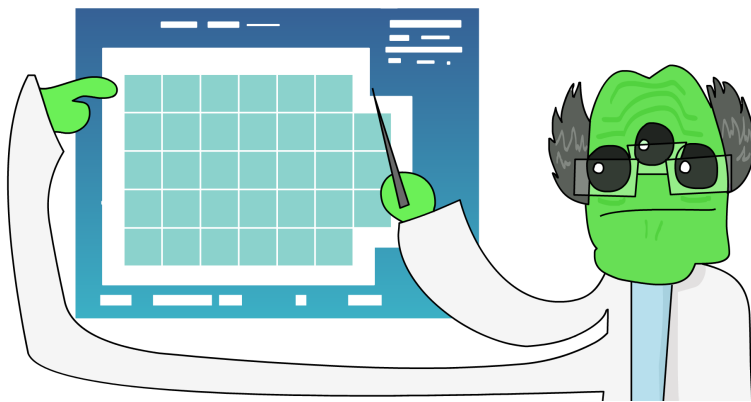
Задача D. Убедить начальника

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Марсианское исследовательское бюро изучает древнюю архитектуру Меркурия.

Младший сотрудник бюро выяснил следующее. По его словам, древние архитекторы Меркурия придерживались строжайших правил. Например, каждое здание представляет собой n квадратных комнат одинакового размера. Для удобства будем считать, что поверхность Меркурия — плоскость, разбитая вертикальными и горизонтальными линиями на единичные квадраты. Тогда каждый квадрат либо занят зданием, либо нет, причем занято всего n квадратов.

Поскольку каждая комната квадратная, то у нее есть четыре стены. Если стена примыкает к другой комнате, то она считается *внутренней*, иначе — *внешней*.



Меркурий славится своими крайне высокими температурами, поэтому внешние стены зданий необходимо делать из специального материала, который не пропускает тепло. На одну внешнюю стену потребуется одна единица материала. Таким образом, чем больше суммарная длина внешних стен здания (т. е. ее периметр), тем больше материала потребуется.

Рассмотрим план здания на рисунке выше. На нем видно, что здания состоит из $n = 33$ комнат, а суммарно у всех комнат 24 внешние стены, т. е. потребуется 24 единицы материала.

Здания построены оптимальным образом, т. е. так, чтобы минимизировать количество материала.

Начальник исследовательского бюро очень увлекается древней архитектурой, поэтому попросил изобразить всевозможные оптимальные планы зданий. Однако его коллеги верно подметили, что таковых может быть очень много, поэтому начальник согласился лишь вычислить лишь их количество и, возможно, хотя бы один оптимальный план (см. раздел «Входные данные»)

Два способа считаются одинаковыми, если они совпадают при наложении друг на друга без поворотов. Таким образом, если повернуть план здания на 90° , то такой новый план может считаться отдельным способом.

Формат входных данных

В первой строке входных данных находится два целых числа t и u ($1 \leq t \leq 2 \cdot 10^5$, $1 \leq u \leq 2$) — количество тестовых примеров и тип задания. Если $u = 1$, то необходимо найти любой способ построить здание оптимальным способом, а если $u = 2$, то необходимо вычислить количество способов это сделать.

Если $u = 2$, то в следующей строке входных данных находится простое целое число m ($10^8 \leq m \leq 10^9 + 9$), по модулю которого необходимо вычислять количество способов.

В каждой из следующих t строк входных данных находится описание тестового примера, состоящее из одного целого числа n ($1 \leq n \leq 4 \cdot 10^5$) — количество комнат, из которых должно состоять здание.

Гарантируется, что если $u = 1$, то сумма n по всем тестовым примерам не превосходит $8 \cdot 10^5$.

Формат выходных данных

Для каждого тестового примера необходимо вывести ответы в формате ниже, разделяя их переводом строки. Формат выходных данных при этом зависит от u во входных данных.

Если $u = 1$, то необходимо в первой строке вывести через пробел два целых числа h и w — высота и ширина участка, на котором должно быть построено здание. Далее, в каждой из следующих h строк необходимо вывести строки s_i , состоящие из w символов «#» и «.» каждая. Если j -й символ строки s_i равен «#», то тогда на соответствующем квадрате должна располагаться комната здания, иначе он считается пустым. Таким образом, мы получаем матрицу из символов. Должно также выполняться условие, что в первой и последней строках матрицы, а также в первом и последнем столбцах, должен быть хотя бы один символ «#», иначе можно было бы вывести такой же план здания, но с меньшими h и w . Если вариантов построить оптимальное здание несколько, выведите любой из них.

Если $u = 2$, то необходимо вывести через пробел два целых числа p и c — количество внешних стен в оптимальном здании, а также остаток от деления количества способов на m .

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$u = 1, t \leq 10, n \leq 10$	4	
2	$u = 1, t \leq 10, n \leq 18$	3	1
3	$u = 1, t, n \leq 100$	8	1 - 2
4	$u = 1, t, n \leq 2000$	5	1 - 3
5	$u = 1$	11	1 - 4
6	$u = 2, t \leq 10, n \leq 10$	5	
7	$u = 2, t \leq 10, n \leq 18$	4	6
8	$u = 2, t, n \leq 100$	12	6 - 7
9	$u = 2, t \leq 10, n \leq 2000$	9	6 - 7
10	$u = 2, t, n \leq 2000$	7	6 - 9
11	$u = 2$, сумма n не превосходит $5 \cdot 10^5$	14	6 - 9
12	$u = 2$	18	6 - 11

Примеры

стандартный ввод	стандартный вывод
3 1 1 2 7	1 1 # 1 2 ## 3 3 . #. ### ###
3 2 1000000007 1 2 7	4 1 6 2 12 22

Замечание

Рассмотрим второй пример.

При $n = 1$ единственный способ построить здание — разместить единственную комнату. При этом периметр будет равен четырем.

При $n = 2$ необходимо разместить две комнаты рядом. Это можно сделать как по вертикали, так и по горизонтали, поэтому есть два способа. Нетрудно убедиться, что в данном случае у здания шесть внешних стен.

При $n = 7$ все 22 оптимальных способа показаны на рисунке ниже:

