Задача A. Сумма дробей с округлением

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Дано целое число n. Найдите значение суммы

$$\left\lfloor \frac{n-1}{1^2} \right\rfloor + \left\lfloor \frac{n-2}{2^2} \right\rfloor + \left\lfloor \frac{n-3}{3^2} \right\rfloor + \ldots + \left\lfloor \frac{2}{(n-2)^2} \right\rfloor + \left\lfloor \frac{1}{(n-1)^2} \right\rfloor.$$

Здесь $\lfloor x \rfloor$ — это округление x вниз: наибольшее целое число, не превосходящее x.

Формат входных данных

В первой строке записано целое число n ($2 \leqslant n \leqslant 10^{18}$).

Формат выходных данных

В первой строке выведите одно целое число — значение суммы.

Примеры

стандартный ввод	*стандартный вывод*
3	2
10	11

Пояснения к примерам

В первом примере $\left\lfloor \frac{2}{1^2} \right\rfloor + \left\lfloor \frac{1}{2^2} \right\rfloor = 2 + 0 = 2$.

Во втором примере

$$\left\lfloor \frac{9}{1^2} \right\rfloor + \left\lfloor \frac{8}{2^2} \right\rfloor + \left\lfloor \frac{7}{3^2} \right\rfloor + \left\lfloor \frac{6}{4^2} \right\rfloor + \left\lfloor \frac{5}{5^2} \right\rfloor + \left\lfloor \frac{4}{6^2} \right\rfloor + \left\lfloor \frac{3}{7^2} \right\rfloor + \left\lfloor \frac{2}{8^2} \right\rfloor + \left\lfloor \frac{1}{9^2} \right\rfloor = 9 + 2 + 0 + \ldots + 0 = 11.$$

Система оценки

В этой задаче два примера и 50 основных тестов. Каждый основной тест оценивается в **2** балла. Как и в других задачах, решение должно правильно работать на примерах — иначе проверка на основных тестах не производится.

Задача B. Половинки

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Даша принесла своей младшей сестре Еве игрушку: детскую книжку «Половинки». В этой книжке две половины: верхняя и нижняя. Страницы в половинах можно листать независимо. Всего в книжке и сверху, и снизу по n разворотов. Развороты в каждой половине пронумерованы целыми числами от 1 до n в порядке, в котором обычно листают книгу.

Сверху на каждом развороте нарисована банка с краской. Все n цветов краски разные, и для удобства пронумерованы целыми числами от 1 до n в том же порядке, что и развороты.

Снизу на каждом развороте нарисован какой-то предмет. Каждый из n предметов имеет один из n цветов, и каждый цвет встречается ровно один раз.

Исходно книжка открыта на первом развороте сверху и первом развороте снизу. За одну секунду Ева может перевернуть одну страницу: либо сверху, либо снизу перейти либо к предыдущему развороту, либо к следующему (конечно, если он есть). Цель игры с книжкой — для каждого цвета k показать одновременно и разворот с краской цвета k сверху, и разворот с предметом цвета k снизу. Цвета можно показывать в любом порядке.

Даша задумалась: как действовать, чтобы достичь цели за минимально возможное время? Помогите ей придумать кратчайшую последовательность действий.

Формат входных данных

В первой строке задано целое число n — количество разворотов в книжке ($2 \leqslant n \leqslant 20$). Во второй строке заданы n целых чисел $a_1, a_2, a_3, \ldots, a_n$ — на каком развороте снизу нарисован предмет первого, второго, третьего, …, n-го цвета ($1 \leqslant a_i \leqslant n$, все числа a_i различны).

Формат выходных данных

Выведите строку, задающую кратчайшую последовательность действий, при которых для каждого из n цветов будет момент, когда сверху открыта банка с краской этого цвета, а снизу — предмет этого цвета. Если кратчайших ответов несколько, выведите любой из них.

Каждое действие записывается одной буквой:

- «`L`» — в верхней половине перевернуть страницу назад,
- «`l`» — в нижней половине перевернуть страницу назад,
- «`R`» — в верхней половине перевернуть страницу вперёд,
- «`r`» — в нижней половине перевернуть страницу вперёд.

Пример

стандартный ввод	**стандартный вывод**
`5` `1 3 5 2 4`	`RrrRRlRrrLLr`

Пояснение к примеру

Для примера запишем в каждую секунду, начиная с нулевой, пару: на каком по счёту развороте открыты верхняя и нижняя половинки. Жирным шрифтом выделены те пары, которые обязательно нужно получить хотя бы один раз.

$$(\mathbf{1}, \mathbf{1})$$
$$\xrightarrow{\texttt{R}} (2,1) \xrightarrow{\texttt{r}} (2,2) \xrightarrow{\texttt{r}} (\mathbf{2}, \mathbf{3})$$
$$\xrightarrow{\texttt{R}} (3,3) \xrightarrow{\texttt{R}} (4,3) \xrightarrow{\texttt{l}} (\mathbf{4}, \mathbf{2})$$
$$\xrightarrow{\texttt{R}} (5,2) \xrightarrow{\texttt{r}} (5,3) \xrightarrow{\texttt{r}} (\mathbf{5}, \mathbf{4})$$
$$\xrightarrow{\texttt{L}} (4,4) \xrightarrow{\texttt{L}} (3,4) \xrightarrow{\texttt{r}} (\mathbf{3}, \mathbf{5})$$

Система оценки

В этой задаче один пример и 50 основных тестов. Каждый основной тест оценивается в **2** балла. Как и в других задачах, решение должно правильно работать на примере — иначе проверка на основных тестах не производится.

Задача C. Многомерные ферзи

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Макс, завоевав рейтинг Эло 932 на популярном шахматном сайте, понял, что классические шахматы себя изжили, и принялся думать над их аналогами. Ему в голову пришёл вопрос: а почему доска для шахмат плоская? Почему бы не поиграть в какую-то более многомерную игру?

Размышляя, он придумал, что в новой версии шахмат доска будет иметь форму d-мерного клетчатого параллелепипеда размером $n_1 \times n_2 \times \ldots \times n_d$, где d натуральных чисел $n_1, \ldots, n_d$ Максу ещё предстоит выбрать. Он решил сделать их такими, чтобы силы различных фигур были как можно более сбалансированными. А какая самая сильная фигура? Конечно, ферзь!

Будем считать, что ферзь на клетке $(q_1, q_2, \ldots, q_d)$ бьёт клетку $(s_1, s_2, \ldots, s_d)$, если существует такое натуральное число r, что вдоль каждой оси $i \in \{1, 2, \ldots, d\}$ выполняется одна из двух возможностей:

- либо вдоль этой оси координаты совпадают, то есть $q_i = s_i$;
- либо вдоль этой оси координаты отличаются ровно на r, то есть $|q_i - s_i| = r$.

При этом вдоль хотя бы одной оси должна выполняться не первая возможность, а вторая, другими словами, клетки $(q_1, q_2, \ldots, q_d)$ и $(s_1, s_2, \ldots, s_d)$ должны быть различны. Это нужно для того, чтобы по правилам ферзь не бил клетку, в которой он и так стоит.

Для $d = 2$ выше описаны стандартные правила перемещения шахматного ферзя. Однако при больших d ферзь, по мнению Макса, становится чересчур мощным: количество клеток, которые может побить ферзь, растёт экспоненциально от d.

Помогите Максу понять, хороши ли будут те или иные размеры доски. Он вам скажет d и все n_i, а также скажет координаты клетки, на которую он поставил ферзя, а вы ответьте, сколько клеток бьёт этот ферзь. Поскольку это число может оказаться слишком большим, выведите его по модулю 998 244 353.

Формат входных данных

В первой строке находится одно целое число d — размерность доски ($1 \leqslant d \leqslant 100\,000$).

Во второй строке находится d целых чисел $n_1, n_2, \ldots, n_d$ через пробел — размеры доски вдоль каждой оси ($1 \leqslant n_i \leqslant 100\,000$).

В третьей строке находится d целых чисел $q_1, q_2, \ldots, q_d$ через пробел — координаты ферзя ($1 \leqslant q_i \leqslant n_i$).

Формат выходных данных

Выведите одно целое число — количество полей на доске $n_1 \times n_2 \times \ldots \times n_d$, которые бьёт ферзь из клетки $(q_1, q_2, \ldots, q_d)$, по модулю 998 244 353.

Система оценки

В этой задаче пять подгрупп тестов. В первой подгруппе $d = 1$, $n_i \leqslant 1000$, она стоит **4** балла. Чтобы их получить, нужно пройти первый тест из условия и все тесты этой подгруппы.

Во второй подгруппе $d = 2$, $n_1 \leqslant n_2 \leqslant 30$, она стоит **10** баллов. Чтобы их получить, нужно пройти первые два теста из условия и все тесты этой подгруппы.

В третьей подгруппе $d \leqslant 6$, $n_i \leqslant 19$, она стоит **10** баллов. Чтобы их получить, нужно пройти первые три теста из условия и все тесты этой подгруппы.

В четвёртой подгруппе $d \leqslant 12$, $n_i \leqslant 1000$, она стоит **38** баллов. Чтобы их получить, нужно пройти все тесты из условия, все тесты предыдущих подгрупп и все тесты этой подгруппы.

В последней подгруппе нет никаких ограничений, не обозначенных в условии и в формате входных данных, она стоит **38** баллов. Чтобы их получить, нужно пройти вообще все тесты: все тесты из условия, все тесты предыдущих подгрупп и все тесты этой подгруппы.

Примеры

стандартный ввод	*стандартный вывод*
1 9 6	8
2 8 8 1 7	21
4 1 2 3 4 1 1 1 1	11
10 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2	59048

Задача D. Собрать многоугольник

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

У вас есть n палок целой длины. Требуется выбрать из них наименьшее возможное количество палок так, чтобы из выбранных палок можно было составить строго выпуклый многоугольник, либо определить, что из имеющихся палок составить строго выпуклый многоугольник невозможно.

Формат входных данных

В первой строке дано целое число n — количество палок ($3 \leqslant n \leqslant 10^5$). Во второй строке даны n целых чисел $a_1, a_2, \ldots, a_n$, разделённые пробелами — длины палок ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите наименьшее количество сторон в строго выпуклом многоугольнике, который можно составить из имеющихся палок. Если составить строго выпуклый многоугольник невозможно, выведите «-1» (без кавычек).

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов, подходящих под ограничения этой группы.

Во всех тестах первой группы $3 \leqslant n \leqslant 4$. За прохождение первой группы можно получить **12** баллов.

В тестах второй группы $3 \leqslant n \leqslant 15$. За вторую группу можно получить ещё **36** баллов.

На тесты третьей группы не накладывается никаких дополнительных ограничений, то есть в ней $3 \leqslant n \leqslant 10^5$. За неё можно получить оставшиеся **52** балла.

Примеры

стандартный ввод	**стандартный вывод**
4 1 2 4 6	4
4 7 4 1 2	-1
4 4 2 5 1	3
3 1 2 1	-1
4 1 1 1 100	3

Пояснения к примерам

В первом примере строго выпуклый многоугольник можно составить, только если использовать все имеющиеся палки.

Во втором примере строго выпуклый многоугольник составить невозможно.

В третьем примере можно составить строго выпуклый многоугольник из всех четырёх палок, но он не наименьший, так как из палок с длинами 4, 2 и 5 можно составить невырожденный треугольник. Поэтому ответ для третьего примера равен 3.

В четвёртом примере всего три палки, и они не образуют невырожденный треугольник, поэтому ответ равен −1.

В пятом примере нельзя построить строго выпуклый многоугольник из всех четырёх палок, но можно из трёх палок длины 1, поэтому ответ равен 3.

Задача E. Минимизация паросочетания

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мебибайта

Ваша задача — по трём целым числам n, Δ и δ построить граф на n вершинах, в котором наибольшая и наименьшая степени вершин равны Δ и δ, соответственно, чтобы размер максимального паросочетания в нём был как можно меньше. Это краткое описание задачи, ниже приведено полное формальное условие.

Пусть G — граф без петель и кратных рёбер. Обозначим через $\delta(G)$ наименьшую степень вершины в нём, $\Delta(G)$ — наибольшую степень вершины в нём, $v(G)$ — количество вершин, $\alpha'(G)$ — размер максимального паросочетания в G, то есть максимальное количество рёбер G, никакие два из которых не имеют общий конец.

Пусть $n > \Delta \geqslant \delta \geqslant 0$ — три целых числа. Обозначим через $\mathcal{G}(n, \Delta, \delta)$ множество всех графов, в которых $v(G) = n$, $\Delta(G) = \Delta$, $\delta(G) = \delta$. Назовём такую тройку целых чисел *хорошей*, если $\mathcal{G}(n, \Delta, \delta) \neq \varnothing$. Обратите внимание, что некоторые тройки не являются хорошими, например, $(2, 1, 0)$ или $(3, 1, 1)$.

Вам дана хорошая тройка (n, Δ, δ), найдите граф $G \in \mathcal{G}(n, \Delta, \delta)$, в котором $\alpha'(G)$ как можно меньше.

Формат входных данных

В единственной строке находится три целых числа n, Δ, δ, образующих хорошую тройку $(0 \leqslant \delta \leqslant \Delta < n \leqslant 400)$.

Формат выходных данных

Выведите целое число m — число рёбер графа. В следующих m строках выведите по два различных целых числа e_i, f_i через пробел — концы очередного ребра $(1 \leqslant e_i, f_i \leqslant n)$. Никакая пара чисел не должна повторяться дважды, даже в обратном порядке.

Система оценки

В этой задаче каждый тест, кроме двух примеров, оценивается независимо вещественным числом от 0 до t_i баллов, где t_i — стоимость этого теста. Оценка вашего решения равняется сумме оценок за каждый тест. Примеры оцениваются в 0 баллов, но на каждый из них должен быть выдан граф с правильным числом вершин, максимальной и минимальной степенью, чтобы решение было принято на проверку. Тесты избраны жюри так, чтобы покрывать как можно больше разнообразных вариантов параметров и соответствующих графов. Всего подготовлено 102 теста. Гарантируется, что тесты 1–2 — это примеры, тесты 3–14 имеют $2 \leqslant n \leqslant 8$, тесты 15–27 имеют $9 \leqslant n \leqslant 16$, тесты 28–40 имеют $17 \leqslant n \leqslant 50$. Стоимость каждого из тестов 3–40 равна $\frac{25}{56} \approx 0.44643$, а стоимость каждого из тестов 41–102 равна $\frac{75}{56} \approx 1.33929$. Эти числа подобраны так, чтобы каждый из тестов 3–40 стоил втрое меньше, чем каждый из тестов 41–102, а суммарная стоимость всех тестов была равна 100 баллам.

Если ваш вывод не описывает граф из $\mathcal{G}(n, \Delta, \delta)$, вы получите вердикт «`Wrong Answer`» или «`Presentation Error`» и 0 баллов за этот тест. Иначе пусть α' — размер максимального паросочетания в вашем ответе, A' — наименьший возможный размер максимального паросочетания. За прохождение теста i вы получите $\frac{1}{\log_2(\alpha' - A' + 1) + 1} \cdot t_i$ баллов, где t_i — максимальное количество баллов за этот тест, а $\log_2$ — двоичный логарифм.

Примеры

стандартный ввод	**стандартный вывод**
`1 0 0`	`0`

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

Пояснения к примерам

В первом примере существует только один граф из $\mathcal{G}(1, 0, 0)$ — граф без рёбер. Если бы тест стоил один балл, то за этот граф вы бы и получили один балл.

Во втором примере существует два неизоморфных графа в $\mathcal{G}(4, 3, 1)$. Один из них — полный двудольный граф $K_{1,3}$, который и приведён в выводе. В нём размер максимального паросочетания равен единице, и это и есть оптимальный ответ. При единичной стоимости теста вы бы получили один балл за данный граф. Второй граф в $\mathcal{G}(4, 3, 1)$ — это треугольник, из которого торчит ребро. В этом графе есть паросочетание размера 2, поэтому из одного балла за тест этот граф дал бы $\frac{1}{\log_2(2-1+1)+1} = \frac{1}{2}$ балла.

Но на самом деле оба примера дают 0 баллов вне зависимости от того, какой граф из $\mathcal{G}(n, \Delta, \delta)$ выведет ваша программа.

Задача F. Здесь был Вася

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вася — уличный художник. Ночью он гуляет по городу с рюкзаком, набитым баллончиками с краской. В укромных местах, увидев однотонную или просто некрасиво раскрашенную стену, Вася останавливается, достаёт краски и рисует на стене что придётся. К большому сожалению Васи, не все одобряют такое творчество, поэтому действовать приходится быстро.

Сейчас Вася нашёл очередную такую стену. На улице горит только один фонарь, поэтому видна только часть стены. Наш герой задумался: нет ли на стене следов его предыдущего рисунка? Если есть, рисовать на ней во второй раз не стоит — но если нет, нужно обязательно оставить ещё один уникальный след на стенах города!

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза.

При первом запуске решение получит схематичное изображение видимой части стены. Стена условно разбита на клетки: пять клеток в высоту и 40 клеток в ширину. При этом фонарь освещает только часть стены шириной в 30 клеток. Каждая клетка покрашена в какой-то один цвет. Цвета обозначаются большими английскими буквами, разные буквы обозначают разные цвета.

Итак, решение получит пять строк, по 30 больших английских букв в каждой. Она уже раскрашена в какие-то цвета, в разных тестах по-разному. Вася ещё не касался этой стены. Нужно раскрасить её — как угодно, но чтобы потом можно было узнать своё творчество. У Васи есть три баллончика с краской: красной, зелёной и синей (обозначаются буквами «`R`», «`G`», «`B`»). Краски каждого цвета хватит, чтобы закрасить этим цветом не более 20 клеток на видимой части стены — независимо от того, какого цвета они были до этого.

После этого могут случиться по порядку три события:

- Придёт Петя — другой уличный художник — и нарисует на этой же стене свою картину — возможно, поверх Васиной. У Пети три баллончика с красками тех же трёх цветов, что и у Васи, но они меньше: каждого хватит не более чем на 10 клеток стены.
- Придёт дворник и закрасит какую-то часть стены шириной в 10 клеток белым цветом (обозначается буквой «`W`»).
- Придёт электрик и перевесит фонарь, в результате чего ночью будет видна другая часть стены шириной в 30 клеток (напомним, что общая ширина стены — 40 клеток).

В каждом тесте все эти действия, как и сам факт их наличия, зафиксированы заранее и не зависят от действий Васи. Петя и дворник могут работать с любой частью стены — даже с той, которая не видна ночью.

После всего этого решение участника будет запущено второй раз. Решение опять получит пять строк, по 30 больших английских букв в каждой — состояние стены после всех действий. В этом запуске нужно понять, что Вася уже рисовал на этой стене, и больше ничего не делать.

Задача состоит в том, чтобы по заданному изображению распознать, первый это запуск на тесте или второй.

Формат входных данных

При каждом запуске решение получает изображение видимой части стены — пять строк, каждая из которых состоит ровно из 30 заглавных английских букв.

Формат выходных данных

Если решение считает, что Вася ещё не рисовал на этой стене — то есть это первый запуск на тесте, — выведите в первой строке «`No`». В следующих пяти строках выведите по 30 символов, задающих рисунок Васи на видимой части стены. Каждый символ может быть точкой («`.`») или одной из трёх букв «`R`», «`G`», «`B`». Точки означают, что клетка останется нетронутой, а буквы говорят о том, что нужно покрасить клетку в соответствующий цвет: красный, зелёный или синий. Каждая буква должна встречаться в рисунке не более 20 раз.

Если же решение считает, что Вася уже рисовал на этой стене — то есть это второй запуск на тесте, — просто выведите в первой строке «`Yes`».

Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Далее показаны два запуска какого-то решения на первом тесте.

стандартный ввод	***стандартный вывод***

`WWWWWWFBRHMVEGNOIJTMKWWWWWWWWW`
`WWWWPLOAALVBZLAMDNCBXKWWWWWWWW`
`WWWWWQZASXDCFGVKBGJNLMXWWWWWWW`
`WWWWWWIWUERPTVEIQLWWWWWWWWWWWW`
`WWWWWWWWWWWWWWWWWWWWWWWWWWWWWW`
	`No`
	`....RRR.......GG......BBB.....`
	`....R..R.....G..G.....B..B....`
	`....RRR......G........BBB.....`
	`....R.R......G.GG.....B..B....`
	`....R..R......GGG.....BBB.....`

`WWWWRRRBRHMVEGGGIJTMKWBBBWWWWW`
`WWWWRLORALVBZGAMGNCBXKBWWBWWWW`
`WWWWRRRASXDCFGVKBGJNLMBBBWWWWW`
`WWWWRWRWUERPTGEGGLWWWWBWWBWWWW`
`WWWWRWWRWWWWWWGGGWWWWWBBBWWWWW`
	`Yes`

Система оценки

Тесты состоят из примера и восьми групп, каждая из которых даёт 12.5 баллов.

В группах 2, 4, 6 и 8 на стене рисует Петя.

В группах 3, 4, 7 и 8 приходит дворник.

В группах 5, 6, 7 и 8 приходит электрик.

Чтобы получить баллы за группу, в которой происходит множество действий S, нужно пройти все тесты, где множество действий является подмножеством S. Например, баллы за группу 7 можно получить, только пройдя все тесты из групп 1, 3, 5 и 7, а также пример из условия. Итоговое количество баллов округляется к ближайшему целому числу (к чётному, если дробная часть равна 0.5).

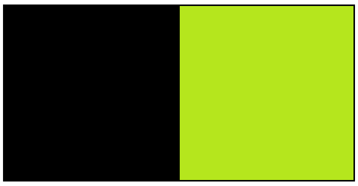
Задача G. Связная фигура

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

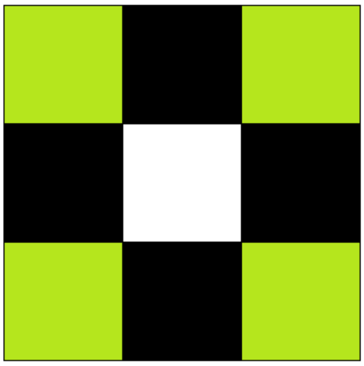
Бесконечный лист клетчатой бумаги раскрашен в шахматном порядке. Какое наименьшее количество белых клеток может быть в связной по стороне клетчатой фигуре, в которой ровно n чёрных клеток?

Формально, клетчатая фигура — это произвольное конечное множество клеток листа. Фигура называется *связной*, если от любой клетки фигуры можно дойти до любой другой, перемещаясь только в соседние по стороне клетки фигуры (выходить за пределы фигуры нельзя).

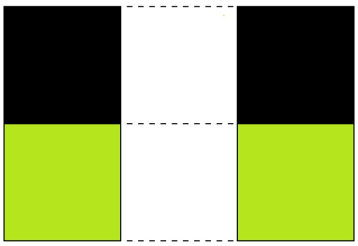
Примеры ниже иллюстрируют, какие фигуры — связные, а какие — нет. В каждом из них для наглядности картинки закрашены только клетки, принадлежащие фигуре, причём белые клетки фигуры закрашены салатовым цветом, чтобы отличаться от белого фона.



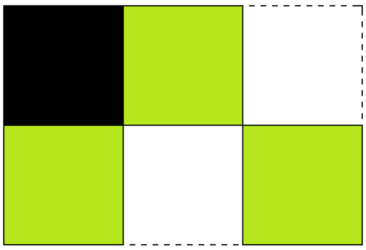
Связная фигура из одной чёрной и одной белой клетки.



Связная фигура с «дыркой». Она состоит из 4 белых и 4 чёрных клеток. Центральная клетка — тоже белая, но не является частью фигуры.



Несвязная фигура — от её левой половины нельзя дойти до правой, оставаясь внутри фигуры.



Всё ещё несвязная фигура, так как ходить можно только между соседними по стороне клетками (касания по углу недостаточно).

Формат входных данных

В единственной строке дано целое число n — количество чёрных клеток в фигуре ($1 \leqslant n \leqslant 10^{18}$).

Формат выходных данных

Выведите одно число — наименьшее возможное количество белых клеток в связной клетчатой фигуре с n чёрными клетками.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов, подходящих под ограничения этой группы.

Во всех тестах первой группы $1 \leqslant n \leqslant 5$. За прохождение всех тестов первой группы можно получить 21 балл.

В тестах второй группы $1 \leqslant n \leqslant 10^5$. За вторую группу можно получить ещё 42 балла.

На тесты третьей группы не накладывается никаких дополнительных ограничений, то есть в ней $1 \leqslant n \leqslant 10^{18}$. За неё можно получить оставшиеся 37 баллов.

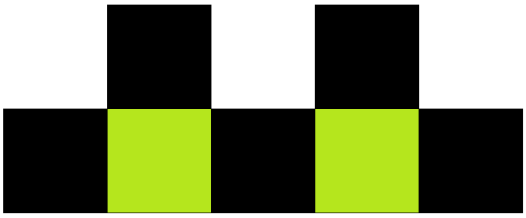
Примеры

стандартный ввод	***стандартный вывод***
1	0
5	2

Пояснения к примерам

В первом примере можно взять фигуру, состоящую из одной чёрной клетки, поэтому ответ на него равен 0.

Можно доказать, что в связной клетчатой фигуре с 5 чёрными клетками всегда будет хотя бы 2 белых клетки. Пример с 2 белыми клетками можно увидеть ниже.



Возможный пример для $n = 5$.

Задача H. Тропический минимум

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Миша получил загадочную посылку из далёкой страны. Открыв её, он обнаружил внутри тропический калькулятор, а также массив целых чисел. При ближайшем рассмотрении Миша понял, что калькулятор умеет вычислять результат довольно длинных выражений с операциями сложения и умножения, однако не умеет обрабатывать скобки. При ещё более детальном рассмотрении он обнаружил, что в качестве результата сложения двух чисел калькулятор возвращает минимальное из чисел вместо их суммы, а в качестве результата умножения — сумму множителей вместо произведения. Мише сразу же стало интересно, какое **минимальное** число может получиться, если он расставит между числами в присланном ему массиве операции сложения и умножения, после чего вычислит на калькуляторе результат полученного выражения. Обратите внимание, что числа в массиве переставлять нельзя, а приоритет умножения у калькулятора выше, чем у сложения, несмотря на то, что сами операции необычны.

Формат входных данных

В первой строке задано одно число T — количество наборов входных данных, для которых нужно решить задачу ($1 \leqslant T \leqslant 10^5$).

В первой строке каждого набора задано число n — размер массива ($1 \leqslant n \leqslant 10^6$). В следующей строке заданы через пробел n целых чисел $a_1, a_2, \ldots, a_n$ — элементы массива ($-2^{31} \leqslant a_i < 2^{31}$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6.

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке одно число — ответ на задачу.

Система оценки

Задача содержит восемь подзадач. Для того, чтобы решение было принято на проверку, необходимо, чтобы оно проходило тест из условия. Баллы за каждую подзадачу будут начислены, если пройдены все тесты этой и всех предыдущих подзадач. Исключения составляют третья подзадача, для прохождения которой не требуется прохождение всех тестов второй подзадачи (но нужно пройти все тесты первой подзадачи), а также последняя подзадача, состоящая из 40 тестов, каждый из которых оценивается индивидуально в 1 балл.

Подзадача	Баллы	Ограничения		
		T	n	a_i
1	10	$T = 1$	$n \leqslant 10$	$\lvert a_i \rvert \leqslant 100$
2	5	$T \leqslant 10^5$	$n \leqslant 10$	$\lvert a_i \rvert \leqslant 100$
3	5	$T \leqslant 10$	$n \leqslant 20$	$\lvert a_i \rvert \leqslant 100$
4	5	$T \leqslant 10$	$n \leqslant 100$	$\lvert a_i \rvert \leqslant 100$
5	5	$T \leqslant 10$	$n \leqslant 100$	$-2^{31} \leqslant a_i < 2^{31}$
6	10	$T \leqslant 10$	$n \leqslant 1000$	$-2^{31} \leqslant a_i < 2^{31}$
7	20	$T = 1$	$n \leqslant 10^4$	$-2^{31} \leqslant a_i < 2^{31}$
8	40	$T \leqslant 10^5$	$n \leqslant 10^6$	$-2^{31} \leqslant a_i < 2^{31}$

Пример

стандартный ввод	*стандартный вывод*
4 7 -1 -1 -1 2 -1 -1 -1 3 7 3 6 5 10 -50 20 -100 30 3 -100 100 100	-4 3 -130 -100

Задача I. Тропический максимум

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Миша получил загадочную посылку из далёкой страны. Открыв её, он обнаружил внутри тропический калькулятор, а также массив целых чисел. При ближайшем рассмотрении Миша понял, что калькулятор умеет вычислять результат довольно длинных выражений с операциями сложения и умножения, однако не умеет обрабатывать скобки. При ещё более детальном рассмотрении он обнаружил, что в качестве результата сложения двух чисел калькулятор возвращает минимальное из чисел вместо их суммы, а в качестве результата умножения — сумму множителей вместо произведения. Мише сразу же стало интересно, какое **максимальное** число может получиться, если он расставит между числами в присланном ему массиве операции сложения и умножения, после чего вычислит на калькуляторе результат полученного выражения. Обратите внимание, что числа в массиве переставлять нельзя, а приоритет умножения у калькулятора выше, чем у сложения, несмотря на то, что сами операции необычны.

Формат входных данных

В первой строке задано одно число T — количество наборов входных данных, для которых нужно решить задачу ($1 \leqslant T \leqslant 10^5$).

В первой строке каждого набора задано число n — размер массива ($1 \leqslant n \leqslant 10^6$). В следующей строке заданы через пробел n целых чисел $a_1, a_2, \ldots, a_n$ — элементы массива ($-2^{31} \leqslant a_i < 2^{31}$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6.

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке одно число — ответ на задачу.

Система оценки

Задача содержит восемь подзадач. Для того, чтобы решение было принято на проверку, необходимо, чтобы оно проходило тест из условия. Баллы за каждую подзадачу будут начислены, если пройдены все тесты этой и всех предыдущих подзадач. Исключения составляют третья подзадача, для прохождения которой не требуется прохождение всех тестов второй подзадачи (но нужно пройти все тесты первой подзадачи), а также последняя подзадача, состоящая из 40 тестов, каждый из которых оценивается индивидуально в 1 балл.

Подзадача	Баллы	Ограничения		
		T	n	a_i
1	10	$T = 1$	$n \leqslant 10$	$\|a_i\| \leqslant 100$
2	5	$T \leqslant 10^5$	$n \leqslant 10$	$\|a_i\| \leqslant 100$
3	5	$T \leqslant 10$	$n \leqslant 20$	$\|a_i\| \leqslant 100$
4	5	$T \leqslant 10$	$n \leqslant 100$	$\|a_i\| \leqslant 100$
5	5	$T \leqslant 10$	$n \leqslant 100$	$-2^{31} \leqslant a_i < 2^{31}$
6	10	$T \leqslant 10$	$n \leqslant 1000$	$-2^{31} \leqslant a_i < 2^{31}$
7	20	$T = 1$	$n \leqslant 10^4$	$-2^{31} \leqslant a_i < 2^{31}$
8	40	$T \leqslant 10^5$	$n \leqslant 10^6$	$-2^{31} \leqslant a_i < 2^{31}$

Пример

стандартный ввод	стандартный вывод
4 7 -1 -1 -1 2 -1 -1 -1 3 7 3 6 5 10 -50 20 -100 30 3 -100 100 100	-1 16 -50 100

Задача J. Раздача подарков

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Скоро Новый Год! Зайчата помогли Деду Морозу упаковать все подарки для детей. Теперь настало время им самим получить подарки!

Всего Деду Морозу помогали n зайчат, пронумерованных целыми числами от 1 до n. Дед Мороз собирается разложить вдоль волшебного коридора n подарков, также пронумерованных по порядку целыми числами от 1 до n. Чтобы получить подарки, зайчата будут по очереди заходить в коридор. Про каждого зайчонка известно, какие из n подарков ему нравятся, а какие нет. Зайчонок прыгает вперёд до **первого** из оставшихся подарков, который ему нравится, забирает его себе и идёт отдыхать. После этого в коридор заходит следующий зайчонок.

Например, пусть $n = 3$, первому зайчонку нравятся все три подарка, второму — только подарок номер 3, а третьему — подарки с номерами 1 и 2. Сначала первый зайчонок возьмёт себе подарок номер 1. Затем второй зайчонок пропустит подарок номер 2 и возьмёт подарок номер 3. Наконец, третий зайчонок возьмёт подарок номер 2 — ему бы понравился и подарок номер 1, но этого подарка в коридоре уже нет.

Увы, может так случиться, что очередной зайчонок зайдёт в коридор, но ни один из оставшихся подарков ему не понравится. Но Дед Мороз на то и волшебник, чтобы этого не допустить!

В этом году Дед Мороз увлёкся алгеброй. Теперь, стоит лишь ему подумать про какую-то квадратную матрицу размера $n \times n$ из нулей и единиц — и подарки окажутся такими, что:

- Если в i-й строке и j-м столбце стоит 1, то i-му зайчонку нравится j-й подарок.
- Если в i-й строке и j-м столбце стоит 0, то i-му зайчонку не нравится j-й подарок.

Квадратная матрица размера $n \times n$ из нулей и единиц называется *счастливой*, если в случае, когда Дед Мороз подумает про эту матрицу, все зайчата получат подарки. Сколько существует различных счастливых матриц? Поскольку их количество может быть очень большим, выведите его остаток от деления на $10^9 + 7$.

Формат входных данных

В единственной строке дано целое число n ($1 \leqslant n \leqslant 10^5$).

Формат выходных данных

Выведите одно число — количество счастливых матриц по модулю $10^9 + 7$.

Система оценки

В этой задаче 22 теста: два примера и 20 скрытых тестов. Если ваше решение проходит оба примера, то за каждый из пройденных скрытых тестов вы получите по 5 баллов.

Гарантируется, что первые 10 скрытых тестов — это все целые числа от 3 до 12.

Примеры

стандартный ввод	**стандартный вывод**
1	1
2	6

Пояснения к примерам

В первом примере подходит только матрица (1).

Во втором примере 6 счастливых матриц: $\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$, $\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}$, $\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}$, $\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}$, $\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$ и $\begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}$.

Матрица $\begin{pmatrix}1 & 1\\1 & 0\end{pmatrix}$ **не** является счастливой: сначала первый зайчонок возьмёт себе первый подарок, затем в коридор зайдёт второй зайчонок — но оставшийся подарок ему не понравится. Также не является счастливой, например, матрица, целиком состоящая из нулей.