

Задача А. Степень матрицы

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

Задана квадратная матрица $n \times n$. Нужно возвести ее в степень t

Формат входных данных

В первой строке задано три целых числа n , t и p ($1 \leq n \leq 100$, $0 \leq t \leq 10^{18}$, $2 \leq p \leq 10^9$).
Далее задана матрица: n строк по n целых чисел. Все числа в матрице неотрицательны и меньше p .

Формат выходных данных

Выведите матрицу: n строк по n чисел, каждое число — остаток от деления элемента на p

Примеры

стандартный ввод	стандартный вывод
3 5 239 1 0 1 1 2 0 3 2 1	120 92 56 102 84 46 21 204 120
5 10 27 1 2 3 4 5 5 4 3 2 1 11 12 13 14 15 15 14 13 12 11 1 11 1 11 1	2 5 7 10 12 16 19 26 2 9 5 9 26 3 20 19 23 18 22 17 12 22 9 19 6

Задача В. Последовательности из 0 и 1 без трех единиц подряд

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

По данному натуральному n определите количество последовательностей длины n из «0» и «1», не содержащих трех единиц подряд.

Формат входных данных

В первой и единственной строке задано число n . $1 \leq n \leq 10^9$.

Формат выходных данных

Выведите ответ по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
3	7

Задача С. Обобщенные числа фибоначчи

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

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано n , найдите значение f_n , взятое по модулю $10^9 + 7$.

Формат входных данных

Неотрицательные целые числа: a, b, c, d, e, n . ($0 \leq a, b, c, d, e \leq 10^9$; $1 \leq n \leq 10^{18}$)

Формат выходных данных

Выведите f_n , взятое по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 1 0 0 0 8	21
1 2 3 4 5 6	775

Задача D. Линейные уравнения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned}a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n\end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \dots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел x_1, \dots, x_n , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333333333 0.666666667

Задача Е. Обращение матрицы.

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

256 мегабайт

Дана квадратная матрица. Найдите обратную к ней.

Формат входных данных

На первой строке входного файла находится одно число N — размер матрицы ($1 \leq N \leq 100$). Далее следуют N строк по N вещественных чисел в каждой — матрица.

Формат выходных данных

Если обратной матрицы не существует, то выведите в выходной файл одну строку NO. Иначе в первой строке выходного файла выведите одно слово YES, а далее выведите N строк по N вещественных чисел в каждой — обратную матрицу. Ответ будет считаться правильным, если абсолютная или относительная погрешность элементов произведения будет не больше, чем 10^{-6}

Примеры

стандартный ввод	стандартный вывод
2 0 1.0 1 0	YES 0.00000000000000000000 1.00000000000000000000 1.00000000000000000000 0.00000000000000000000
2 0 0 1 0	NO

Задача F. Добавление векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В изначально пустое множество один за другим добавляются m битовых векторов размера n . После добавления каждого скажите, можно ли его представить как хог векторов, добавленных до него.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n \leq 50$, $1 \leq m \leq 10000$). В следующих m строках записаны вектора.

Формат выходных данных

Для каждого вектора выведите 1, если его можно представить как хог предыдущих и 0 если нельзя.

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	0 0
3 4 100 111 011 010	0 0 1 0
3 4 000 111 111 111	1 0 1 1

Задача G. Всё, везде и Гаусс

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

Для множества неотрицательных целых чисел S назовем $X(S)$ множество всех возможных XOR-значений, которые можно получить, взяв XOR любого подмножества S :

$$X(S) = \{x_1 \oplus x_2 \oplus \dots \oplus x_k \mid x_1, \dots, x_k \in S\},$$

где \oplus обозначает операцию побитового исключающего ИЛИ. Допускается подмножество из нуля элементов, и XOR его элементов это 0, поэтому $0 \in X(S)$ всегда.

Требуется поддерживать множество S и обрабатывать запросы четырёх типов:

- `add x` — добавить число x в S ;
- `can x` — проверить $x \in X(S)$;
- `cnt` — вывести количество различных значений в $X(S)$;
- `max` — вывести максимальный элемент множества $X(S)$.

Изначально S пусто.

Формат входных данных

В первой строке входных данных дано одно число Q ($1 \leq Q \leq 10^5$) — количество запросов. В каждой из следующих Q строк вводится запрос одного из четырех типов.

- `add x` — добавить число x в S ($0 \leq x < 2^{62}$);
- `can x` — проверить $x \in X(S)$ ($0 \leq x < 2^{62}$);
- `cnt` — найти $|X(S)|$;
- `max` — найти максимальное число в $X(S)$.

Формат выходных данных

Для каждого запроса, кроме `add`, необходимо вывести ответ:

- `can x`: вывести 1, если $x \in X(S)$, иначе 0;
- `cnt`: вывести $|X(S)|$;
- `max`: вывести максимальный элемент из $X(S)$.

Примеры

стандартный ввод	стандартный вывод
6 add 11 add 3 max add 38 max cnt	11 46 8
24 max cnt can 0 can 1 add 9 max cnt can 12 can 3 add 5 can 12 max cnt add 12 max cnt can 4 add 8 can 4 can 1 can 0 can 2 cnt max	0 1 1 0 9 2 0 0 1 12 4 12 4 0 1 1 1 0 8 13

Задача Н. Симпатичные узоры наносят ответный удар

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся два натуральных числа n и m . $1 \leq n \leq 10^{100}$, $1 \leq m \leq 5$, $1 \leq p \leq 10000$.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$ по модулю p . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Примеры

стандартный ввод	стандартный вывод
2 2 20	14
3 3 7	0

Задача I. Кузнечик

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

Кузнечик двигается из клетки 0 в клетку n . За один прыжок он может прыгать на 1, 2 или 3 клетки вперед. Ваша задача — посчитать, сколько способов допрыгать у него есть. Ситуация усложняется тем, что иногда некоторые клетки становятся непроходимыми, а иногда наоборот. Вам нужно быстро пересчитывать число способов добраться из 0 в n после каждого изменения.

Формат входных данных

Первая строка содержит два числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) — число клеток и число изменений. Следующие m строк содержат описания изменений. Каждое изменение описывается числом x ($1 \leq x \leq n - 1$), номером клетки, для которой изменился статус (то есть если она была проходимой, то стала непроходимой и наоборот).

Формат выходных данных

Выведите $m + 1$ число — число способов добраться из 0 до n в начале и после каждого изменения по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 4 1 2 1 2	4 2 1 2 4
100 2 33 56	347873931 85092847 684577937

Задача J. Деревья и XOR-запросы снова

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 6.5 секунд
Ограничение по памяти: 512 мегабайт

Вам дано дерево, состоящее из n вершин. На каждой вершине написано целое число; на i -й вершине написано целое число a_i .

Вам необходимо обработать q запросов. i -й запрос состоит из трех целых чисел x_i , y_i и k_i . Для этого запроса вам нужно ответить, возможно ли выбрать набор вершин v_1, v_2, \dots, v_m (возможно, пустой) так, чтобы:

- каждая вершина v_j находится на простом пути между x_i и y_i (концы также могут быть использованы);
- $a_{v_1} \oplus a_{v_2} \oplus \dots \oplus a_{v_m} = k_i$, где \oplus обозначает побитовый оператор XOR.

Формат входных данных

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

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 2^{20} - 1$).

Затем следуют $n - 1$ строк. Каждая из них содержит два целых числа u и v ($1 \leq u, v \leq n$; $u \neq v$), обозначающие ребро дерева.

Следующая строка содержит одно целое число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

Затем следуют q строк. i -я из них содержит три целых числа x_i , y_i и k_i ($1 \leq x_i, y_i \leq n$; $0 \leq k_i \leq 2^{20} - 1$).

Формат выходных данных

Для каждого запроса выведите YES, если возможно сформировать набор вершин, удовлетворяющий условиям. В противном случае выведите NO.

Вы можете печатать каждую букву в любом регистре

Пример

стандартный ввод	стандартный вывод
4	YES
0 1 2 10	YES
2 1	NO
3 2	YES
4 2	YES
8	NO
3 3 0	YES
3 4 1	YES
3 4 7	
1 3 1	
1 3 2	
1 3 10	
1 4 10	
1 4 11	

Задача К. ЛКШатская ПСП

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

Как известно, в ЛКШ смотрят на вещи под самыми разными углами, поэтому и правильные скобочные последовательности они задают нестандартным способом:

- Пустая скобочная последовательность считается правильной.
- Если S считается правильной, то $)S($, (S) , $[S]$, $]S[$, $\{S\}$, $\}S\{$, $\langle S \rangle$ и $\rangle S \langle$ тоже считаются правильными.
- Если S и T считаются правильными, то $S + T$ тоже считается правильной (здесь $+$ означает конкатенацию строк).

Мальчик Вася решил посетить ЛКШ. Но вот беда, для этого надо пройти ЛКШатский тест на интеллект! В самом сложном задании теста даётся строка s , состоящая из скобок, и к ней даются m заданий двух видов:

1. Заменить скобку на позиции a_i .
2. Сказать, считается ли подстрока s на позициях с l_i по позицию r_i включительно правильной скобочной последовательностью в ЛКШ.

Вася очень просит вас помочь ему пройти тест.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 200\,000$) — длина скобочной последовательности.

Во второй строке содержится строка s длины n , состоящая из скобок $() [] \{ \} \langle \rangle$ — исходная строка, данная Васе.

В третьей строке содержится целое число m ($1 \leq m \leq 200\,000$) — количество заданий теста.

В следующих m строках заданы запросы. В i -й из следующих строк в начале содержится целое число t_i ($1 \leq t_i \leq 2$).

- Если $t_i = 1$, то далее строка содержит целое число a_i и символ c_i ($1 \leq a_i \leq n$). В этом случае требуется в строке s на позиции a_i заменить скобку на c_i . Гарантируется, что c_i является одной из скобок $() [] \{ \} \langle \rangle$.
- Если $t_i = 2$, то далее строка содержит два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$). В этом случае требуется узнать, считается ли подстрока s на позициях с l_i по позицию r_i правильной скобочной последовательностью в ЛКШ.

Формат выходных данных

Для каждого запроса второго типа выведите **Yes** (без кавычек), если скобочная последовательность считается правильной, и **No** (без кавычек) иначе.

Примеры

стандартный ввод	стандартный вывод
6)()((7 2 1 6 1 4) 2 2 5 1 3 [1 4] 2 1 6 2 4 5	Yes Yes Yes No
10 >())(][<}{ 6 2 1 10 1 3 (2 1 10 2 2 5 1 2) 2 1 10	Yes No No Yes

Замечание

В первом примере:

1. В первом задании просят сказать, считается ли подстрока `)()((` правильной. Ответ **Yes**, так как `)` и `(` считаются правильными, а эта подстрока представляется как сумма таких строк.
2. Во втором задании просят заменить скобку на 4-й позиции на `)`. После этого строка будет равна `)()()`.
3. В третьем задании просят сказать, считается ли подстрока `()(` правильной. Ответ **Yes**.
4. В четвёртом задании просят заменить скобку на 3-й позиции на `[`. После этого строка будет равна `)([())`.
5. В пятом задании просят заменить скобку на 4-й позиции на `]`. После этого строка будет равна `)(]()`.
6. В шестом задании просят сказать, считается ли подстрока `)([())` правильной. Ответ **Yes**.
7. В седьмом задании просят сказать, считается ли подстрока `]()` правильной. Ответ **No**.

Во втором примере:

1. В первом задании просят сказать, считается ли подстрока `>())(][<` правильной. Ответ **Yes**, так как `()` и `)(` считаются правильными, поэтому их сумма `()(` считается правильной, поэтому строка `>())(<` считается правильной. Также `}{` считается правильной, поэтому исходная подстрока считается правильной.
2. Во втором задании просят заменить скобку на 3-й позиции на `(`. После этого строка будет равна `>())(][<`.
3. В третьем задании просят сказать, считается ли подстрока `>())(][<` правильной. Ответ **No**, так как иначе строка `)(()` считалась бы правильной, но нетрудно убедиться, что это не так.

4. В четвёртом задании просят сказать, считается ли подстрока $((() ($ правильной. Нетрудно убедиться, что правильной она не считается, поэтому ответ **No**.
5. В пятом задании просят заменить скобку на 2-й позиции на $)$. После этого строка будет равна $>) () [] [<$.
6. В шестом задании просят сказать, считается ли подстрока $>) () [] [<$ правильной. Ответ **Yes**, так как $) () ($ правильная, поэтому и $>) () (<$ правильная, поэтому и исходная подстрока правильная.