

A. A sad number

Ввиду некоторых печальных событий, никак не связанных с учёбой в университете, Денис считает цифру 2 самой грустной цифрой. Соответственно натуральные числа, в десятичной записи которых есть цифра 2, Денис называет «грустными» числами; числам же, которые можно разложить на сумму двух «грустных» чисел, Денис дал название «очень грустные» числа. Поскольку определение вышло слишком сложным, помогите Денису определить, является ли данное число «очень грустным» или нет.

Ввод.

Одно целое число n , где $1 \leq n \leq 10^{18}$.

Вывод.

Слово YES, если введенное число является «очень грустным», и NO — иначе.

Пример.

Ввод	Вывод
10	NO
14	YES
3018	YES

Комментарий.

Число 10 нельзя разложить на сумму двух грустных чисел.

Число $14 = 12 + 2$, здесь и 2, и 12 содержат двойку в своей записи.

Число $3018 = 2016 + 1002$ — оба слагаемых являются грустными числами.

В. Brute force

Тимур считает, что все задачи по программированию можно решить перебором. Казалось бы, чего сложного, например, перебрать все возможные пары различных натуральных чисел (a, b) не более некоторого n и вывести максимальный НОД из полученных? Не правда ли?

Ввод.

Одно целое число n , где $2 \leq n \leq 10^{18}$.

Вывод.

Одно целое число — максимальный НОД чисел a и b при всех возможных $1 \leq a < b \leq n$.

Пример.

Ввод	Вывод
2	1
5	2

Комментарий.

Во втором примере все возможные НОДы равны:

$\text{НОД}(1, 2) = 1$, $\text{НОД}(1, 3) = 1$, $\text{НОД}(1, 4) = 1$, $\text{НОД}(1, 5) = 1$, $\text{НОД}(2, 3) = 1$, $\text{НОД}(2, 4) = 2$,
 $\text{НОД}(2, 5) = 1$, $\text{НОД}(3, 4) = 1$, $\text{НОД}(3, 5) = 1$, $\text{НОД}(4, 5) = 1$.

Максимальным из полученных чисел является 2.

C. Curtains

Адиль хочет применить полученные знания на практике после окончания университета. Чтобы стать ближе к проектированию «умного дома», Адиль уже сейчас пытается написать что-то своё. Его первый проект — автоматическое управление шторами. Деталей разработки своего проекта Адиль не раскрывает, но на данный момент его у него имеется N петелек для штор, которые крепятся к карнизу в точках a_1, a_2, \dots, a_N (координаты в см $1 \leq a_1 < a_2 < \dots < a_N \leq 10^5$). А его «умный дом» может выполнять запросы вида:

1. сдвинуть k -ю петельку вправо или влево на 1 см;
2. сдвинуть k -ю петельку вправо или влево на 1 см.

При этом совпадать петельки в процессе действий не могут — такие действия следует игнорировать. Проект ещё далёк от завершения, а Адилью нужно готовиться к сессии. Чтобы контролировать, насколько сильно провисают шторы, Адилью важно знать максимальное расстояние между соседними петельками после каждого запроса. Обратите внимание, что карниз можно считать бесконечным и координаты могут становиться отрицательными в процессе перемещений.

Ввод.

В первой строке целое число N — число штор, где $2 \leq N \leq 10^5$,
 во второй строке N целых чисел a_1, a_2, \dots, a_N — начальное положение штор, где $1 \leq a_i \leq 10^5$,
 в третьей строке целое число Q — число запросов, где $1 \leq Q \leq 10^5$,
 в следующих Q строках содержится описания запросов одного из двух видов:
 «R k » соответствует запросу типа «сдвинуть k -ую петельку вправо на 1 см»;
 «L k » соответствует запросу типа «сдвинуть k -ую петельку влево на 1 см».

Вывод.

Q целых чисел в столбик — максимальное расстояние между соседними шторами после каждого запроса.

Пример.

Ввод	Вывод
3	5
1 5 10	5
3	4
R 2	
L 3	
R 1	
2	1
1 2	1
3	2
R 1	
L 2	
R 2	

D. Dimitriy and broken sum

Димитрий отлично знает, как находить сумму членов арифметической прогрессии, поэтому на этот раз ему дали задачу сложнее. Необходимо найти сумму $a_0 + a_1 + a_2 + \dots + a_n$, где a_i отличается от i только k -ым битом справа в своей двоичной записи. Если число содержит недостаточно двоичных цифр, следует добавить незначащие нули слева.

Ввод.

Два целых числа n и k , где $0 \leq n \leq 10^9$ и $1 \leq k \leq 30$.

Вывод.

Одно целое число — ответ на задачу.

Пример.

Ввод	Вывод
7 3	28
5 3	23

Комментарий.

При $k = 3$ последовательность $\{a_n\}$ будет выглядеть следующим образом:
4, 5, 6, 7, 0, 1, 2, 3, 12, 13, 14, ...

Е. Excursion

Ребята из команды Snowy Cube сейчас учатся на 3 курсе ВМК и проходят один из самых интересных предметов — “Машинная графика”, где они программируют отрисовку трёхмерных изображений с помощью библиотеки OpenGL. Как и полагается, первой демонстрационной программой является отрисовка куба. Ребята так хотят научиться писать движки для трехмерных игр, что уже начали разрабатывать трехмерную игру «Экскурсия по снежному кубу». Изначально игровое пространство состоит из куба размера A , ребра которого параллельны осям координат. Суть игры заключается в том, чтобы добраться из угла куба $(0, 0, 0)$ в противоположный угол с координатами (A, A, A) . Двигаться можно только прыжками по снежинкам, расставленным внутри куба, при этом общая длина всех прыжков не должна превышать данного параметра K (уровень сложности игры). Штраф в игре вычисляется как длина максимального прыжка, совершенного во время преодоления пути. Ясно, что игроки будут пытаться получить минимальный штраф. Чему будет равен абсолютный рекорд для данной карты?

Ввод.

В первой строке два целых числа A и K — размер куба и уровень сложности ($1 \leq A \leq 1000$, $1 \leq K \leq 2000$);

во второй строке — число снежинок N на данной карте ($0 \leq N \leq 100$);

каждая из следующих N строк содержит три целых числа x_i , y_i , z_i — координаты i -ой снежинки ($0 \leq x_i, y_i, z_i \leq A$).

Вывод.

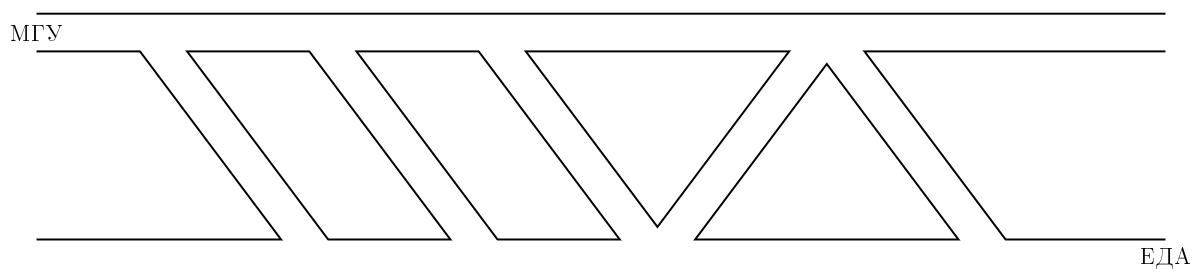
Одно вещественное число — минимальный штраф L (с точностью в 2 знака после запятой), с которым можно пройти данную карту. Если эту карту на данном уровне сложности пройти невозможно, то вывести -1.

Пример.

Ввод	Вывод
10 20 5 5 0 0 10 0 0 10 5 0 10 10 0 10 10 5	17.320
10 30 5 5 0 0 10 0 0 10 5 0 10 10 0 10 10 5	7.071
10 40 5 5 0 0 10 0 0 10 5 0 10 10 0 10 10 5	5.000
10 16 2 6 2 1 4 0 6	-1

F. Food getting ways

Знаете ли вы, где можно вкусно перекусить рядом с филиалом МГУ? В соседнем с нашим архитектурно-строительном корпусе работает замечательная столовая с оригинальным названием The Stolovka. От нашего корпуса до корпуса, в котором находится The Stolovka, можно пройти по одной из двух параллельных аллей, соединённых между собой некоторой системой из попарно не пересекающихся переулков. Ерулан, как большой любитель покушать, собрался посетить данное место. Но, кроме прочего, он еще и большой любитель математики, поэтому он решил воспользоваться случаем и на основе этого маленького приключения составить задачу.



Ерулан записал её условие в следующем формальном виде: две аллеи — горизонтальные отрезки. Выходит он из верхнего левого угла, прийти должен в правый нижний. Переулки между аллеями бывают 2 видов: / и \. Ерулан всегда движется слева направо. На каждом шаге он может либо продолжить движение по той же аллее, либо перейти по переулку на другую, если есть такая возможность. Посчитайте количество способов попасть в столовую.

Ввод.

Одно целое число N , где $1 \leq N \leq 10000$.

На следующей строке строка длины N из символов вида '/' и '\'.

Вывод.

Одно целое число — ответ по модулю $10^9 + 7$.

Пример.

Ввод	Вывод
5 \\\/\	7
2 //	0

Комментарий.

обратите внимание, что в языке C и C++ константы прямого и обратного слеша обозначаются как '/' и '\\'.

G. Great and Mighty

Илья учится уже на 4 курсе, но всё ещё отлично помнит что, студенты ВМК за первые три семестра знакомятся с тремя не самыми популярными, но одними из самых важных для понимания основ, языками программирования: pascal, pasm и C. Правда, эта информация никак не помогла ему во время сдачи уравнений математической физики, и навряд ли поможет вам.

Ввод.

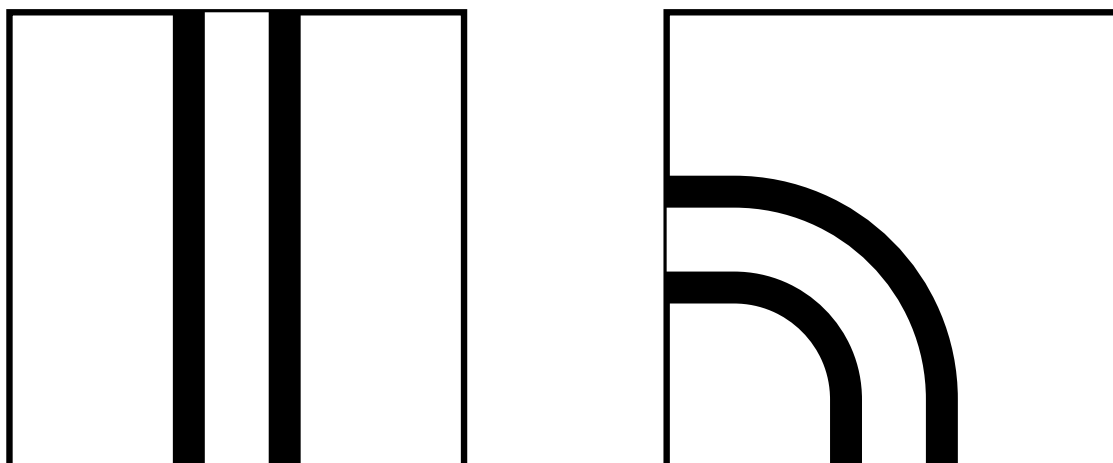
Одно целое число от 1 до 10.

Пример.

Ввод	Вывод
9	6 5 4
2	3 3 3

Н. Hobby

Темирхан очень обрадовался не столько победе на личной олимпиаде в прошлом году (которая проводится традиционно в декабре в филиале МГУ), сколько подарку в виде сборной модели здания МГУ. Он увлекся различным конструированием, и в частности, задумался о сборке железной дороги. Для этого Темирхан купил себе большую пачку деталей двух видов: прямой прогон (длины 2) или поворот на 90 (ширина 1, высота 1).



После того, как он принес A деталей первого вида и B деталей второго вида, у него моментально возник вопрос: можно ли составить замкнутую железную дорогу, используя все детали? Помогите ему ответить на этот странный вопрос.

Ввод.

Два целых числа A и B , где $0 \leq A \leq 10^9$, $0 \leq B \leq 10^9$ и $A + B > 0$.

Вывод.

Слово YES, если собрать дорогу возможно, и NO — иначе.

Пример.

Ввод	Вывод
0 4	YES
0 8	NO
3 3	NO

Комментарий.

Дорога не может перекрываться, поэтому 8 уголков не подходят.

I. IMC problem

Надира и Саша этим летом участвовали в IMC — международной студенческой олимпиаде по математике в Болгарии. На тренировках они решили огромное количество различных задач, в том числе и по теории чисел. Иногда попадались задачи, которые проще запрограммировать, чем решить честно. Например, как вам такая задача:

найти n -ое число вида $2^p \cdot 5^q$, то есть, n -ое число последовательности 1, 2, 4, 5, 8, 10, 16, 20, 25, ...

Ввод.

Одно положительное целое число n такое, что ответ гарантированно не превосходит 10^{18} .

Вывод.

Одно целое число — n -ое число последовательности.

Пример.

Ввод	Вывод
8	20
100	1000000
500	1000000000000000

J. Join the knowledge

Куат изучает очень много математики. За первые два семестра он прослушал курсы «Алгебра», «Аналитическая геометрия», «Линейная алгебра» и «Математический анализ». Знания по всем этим предметам занимают большую часть его памяти. Свою память он представляет в виде таблицы $n \times n$. Если клетка памяти занята знаниями, то в таблице это отмечено точкой. Если она еще свободна — звездочкой. Будучи настоящим математиком, Куат верит, что все предметы и знания по ним должны быть связаны друг с другом (две единицы знаний считаются соседними, если они имеют общую сторону; все единицы считаются связанными между собой, если для любых двух единиц знания A и B можно составить цепочку $A = v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k = B$, где v_i и v_{i+1} — соседние клетки для любого i). Куат очень надеется, что его знания связаны друг с другом. А если нет, то он готов потратить еще один вечер и заполнить знаниям еще один кусок памяти, чтобы они стали связаны. Гарантируется, что у Куата уже есть как минимум две единицы знаний. Сможет ли он добиться того, чтобы все его знания стали связаны после добавления одной единицы знаний?

Ввод.

В первой строке одно целое число n , где $2 \leq N \leq 300$, — размер матрицы знаний. На следующих n строках описание таблицы знаний $n \times n$ из символов $*$ (звездочка) и $.$ (точка).

Вывод.

Слово YES, если Куат может добиться связности знаний, заменив не более одной звездочки на точку, и NO — иначе.

Пример.

Ввод	Вывод
3 *** .*. .*.	YES
3 *.* .*. *.*	YES
3 .*. *.* .*.	NO

К. Keg and dipper

В Казахстанском филиале МГУ с недавнего времени появилась традиция пить чай на тренировках по программированию. Команда Big dipper сразу начала навязывать свои привычки: всё мерить в ковшах, а чай пить только из бесконечно большого бочонка. Ещё ребята определили комфортную температуру чая, при которой его можно пить: от A до B градусов включительно. Кипяток из чайника имеет температуру H , молоко — температуру C . Количество молока и кипятка не ограничено. Помогите им выяснить, какое минимальное количество мерительных ковшей необходимо налить в бочонок, чтобы получить в нём чай комфортной температуры?

Ввод.

В первой строке два целых числа A и B .
Во второй строке два целых числа C и H , где $0 \leq C < A < B < H \leq 10^6$.

Вывод.

Одно целое число — минимальное число ковшей, при котором можно получить комфортную температуру чая.

Пример.

Ввод	Вывод
36 37 25 95	6
23 24 20 30	3

Комментарий.

Температура чая, получаемой при смешивании x ковшей кипятка и y ковшей молока, равна $T = \frac{x \cdot H + y \cdot C}{x + y}$.
В первом примере, если взять 5 ковшей кипятка и 1 ковш молока, то получим температуру $36\frac{2}{3}$, которая является комфортной.
В втором примере, если взять 2 ковша кипятка и 1 ковш молока, то получим температуру $23\frac{1}{3}$, которая является комфортной.

L. Lazy programming

На механико-математическом факультете филиала МГУ студенты 1 курса проходят замечательные предметы «Элементы теории чисел» и «Технология программирования на ЭВМ». Айтмухаммед и Павел в ответ на «ленивые математические вычисления», про которые услышали на программировании, решили изобрести метод «ленивого программирования» для задач по теории чисел. На такой шаг они пошли с целью не увеличивать всемирную энтропию работающих компьютеров, на которых написаны «решения в лоб», выполняющие много лишних циклов. Для этого они выводят формулу для решения задачи на бумаге так, чтобы в программе не было циклов вообще. Вот задача, которую они выбрали в качестве первого претендента:

*проверить, является ли чётным число $\tau(1) + \tau(2) + \tau(3) + \dots + \tau(n)$,
где $\tau(k)$ — количество натуральных делителей числа k .*

Попробуйте написать эту программу и Вы.

Ввод.

Одно целое число n , где $1 \leq n \leq 10^{18}$

Вывод.

Слово YES, если результат вычислений будет четным и NO — иначе.

Пример.

Ввод	Вывод
5	YES
10	NO