Задача A. Абсолютные построения

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

В классе N детей, причём у любых двух детей рост различается.

Перед уроком физкультуры класс выстроился в одну линию. Физрук Сергей Богданович определяет *расхлябанность* построения как количество пар детей (не обязательно стоящих рядом), в которых более высокий стоит левее более низкого.

Назовём построение *абсолютным*, если после того, как любые двое детей, стоящие рядом, однократно поменяются местами, расхлябанность либо **всегда** увеличится, либо **всегда** уменьшится.

По заданному N определить количество абсолютных построений.

Формат входных данных

Первая строка входных данных содержит одно целое число N ($2 \leqslant N \leqslant 10^4$) — количество детей.

Формат выходных данных

Выведите одно целое число — количество абсолютных построений.

Пример

стандартный ввод	стандартный вывод
42	2

Задача B. Больше финалистов!

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

Правила отбора студенческих команд на Moscow Regional Contest из квалификационного раунда формулируются следующим образом:

1. Первые 25 команд выходят в следующий этап.
2. Лучшая команда каждого университета, если она ещё не вышла по правилу 1 и решила как минимум одну задачу, выходит в следующий этап.
3. Из числа команд, не вышедших по правилам 1 и 2, представляющих университет, от которого вышло менее 4 команд по правилу 1, и занявших по своему университету 2-4 места, 25 команд выходят в следующий этап (если они решили хотя бы одну задачу).

Известно, что в квалификационном раунде приняли участие команды из $N \geqslant 24$ университетов, от каждого университета участвовало минимум четыре команды, все команды решили хотя бы одну задачу. Определите максимальное и минимальное количество команд, которые по этим правилам выходят в следующий этап.

Формат входных данных

Первая строка входных данных содержит одно целое число N — количество университетов в отборе ($24 \leqslant N \leqslant 110$).

Формат выходных данных

Выведите два целых числа — наибольшее и наименьшее количество команд, которые могут выйти в следующий этап при описанных выше условиях.

Пример

стандартный ввод	стандартный вывод
27	52 76

Задача C. Выпуклый многоугольник

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

Как известно из элементарной геометрии, внутренние углы выпуклого многоугольника должны быть строго больше 0 и строго меньше 180 градусов, а их сумма должна в точности быть равна $180 \cdot (n - 2)$, где $n \geqslant 3$ — количество вершин многоугольника.

Вам даны k целых углов в интервале от 1 до 179 градусов включительно. Выясните, существует ли выпуклый многоугольник, в котором **все** углы задаются целым числом градусов, причём углы при каких-то k подряд идущих вершинах совпадают с заданными. Если такой многоугольник существует, выведите наименьшее и наибольшее возможное количество вершин многоугольника.

Формат входных данных

Первая строка входных данных содержит одно целое число k ($1 \leqslant k \leqslant 1000$). i-я из последующих k строк содержит одно целое число a_i ($1 \leqslant a_i \leqslant 179$) — величину i-го угла в градусах.

Формат выходных данных

Если выпуклого многоугольника с требуемыми свойствами не существует, выведите -1. В противном случае выведите два целых числа — наибольшее и наименьшее количество вершин многоугольника, углы которого задаются целым числом градусов, и углы при k подряд идущих вершинах совпадают с заданными.

Примеры

стандартный ввод	стандартный вывод
3 90 90 90	4 93
1 60	3 241
3 1 1 1	-1

Задача D. Геометрическая подпоследовательность

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

Геометрическая прогрессия G задаётся следующим образом: $G_0 = q$, $G_{i+1} = G_i \cdot r$, где $q > 0$ и $r > 1$ — целые числа.

Арифметическая прогрессия A задаётся следующим образом: $A_0 = b$, $A_{i+1} = A_i + d$, где $a, d > 0$ — целые числа. d называется *разностью* арифметической прогрессии.

Назовём геометрическую прогрессию *вложенной* в арифметическую прогрессию A, если **все** члены прогрессии G являются и членами прогрессии A. Например, всякая геометрическая прогрессия с целыми q и r вложена в арифметическую прогрессию с $a = 1$ и $d = 1$ (которая включает в себя все целые числа).

По заданным q и r определите, сколько различных уникальных целых значений может принимать разность арифметической прогрессии A, в которую вложена геометрическая прогрессия, заданная q и r.

Формат входных данных

Первая строка входных данных содержит целое число q ($1 \leqslant q \leqslant 10^6$). Вторая строка входных данных содержит целое число r ($2 \leqslant r \leqslant 10^6$).

Формат выходных данных

Выведите одно целое число — количество уникальных целых значений, которые может принимать разность арифметической прогрессии, в которую вложена геометрическая прогрессия, задаваемая q и r.

Пример

стандартный ввод	стандартный вывод
1 7	4

Задача E. Диаметры

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

Сеть железнодорожных диаметров в городе M состоит из станций. Некоторые пары станций соединены перегонами с двусторонним движением. Одна пара станций может быть соединена напрямую не более, чем одним перегоном, между любыми двумя станциями можно проехать по одному или нескольким перегонам.

- Если станция соединена перегоном ровно с одной станцией (пусть это станция a), такая станция является *конечной*. Все прибывшие пассажиры на этой станции **обязаны** выйти из поезда, после чего поезд направляется в тупик. Под посадку в направлении станции a подаётся пустой поезд из тупика.
- Если станция соединена перегоном ровно с двумя станциями (например, a и b), такая станция является *обычной*. В каждом из направлений поезд останавливается на этой станции, после чего продолжает движение дальше (то есть в одном направлении приезжает со стороны станции a, останавливается для посадки и высадки, продолжает движение к станции b, в другом направлении приезжает со стороны станции b, останавливается для посадки и высадки, продолжает движение к станции a).
- Если станция соединена перегоном более, чем с двумя станциями, такая станция является *узловой*. На узловой станции можно перейти с любого из перегонов на любой другой и произвольным образом выбрать направление.

Архитектурно каждая станция может иметь два типа:

- Если станция является *островной* (тип 1), то платформа находится посередине между путями, и пассажир может свободно выйти из поезда, едущего в одном направлении, и пересесть на поезд, идущий в обратном направлении, даже если станция не является узловой.
- Если станция является *береговой* (тип 2), то пути идут между платформами, и пассажир для перехода в обратном направлении на неузловой станции должен выйти с одной платформы, завершая оплаченную поездку, и зайти на станцию с другой, оплачивая новую поездку. Отметим, что на узловых станциях бесплатные переходы между направлениями возможны **вне зависимости** от архитектуры станции.

По заданному списку станций (с информацией о типе каждой станции) и конфигурации перегонов определите, какое наибольшее количество станций можно посетить, оплатив поездку ровно один раз. Станции посадки и высадки могут быть выбраны произвольно. Разрешается повторно проезжать одни и те же станции и даже одни и те же перегоны.

Формат входных данных

Первая строка входных данных содержит два целых числа N и M — количество станций и количество перегонов, соответственно. Станции занумерованы последовательными целыми числами от 1 до N ($2 \leqslant N \leqslant 10^5$, $1 \leqslant M \leqslant \min(2 \cdot 10^5, \frac{N(N-1)}{2})$).

Вторая строка содержит N целых чисел a_i ($1 \leqslant a_i \leqslant 2$). i-е число равно 1, если i-я станция имеет островной тип, и 2, если береговой.

Далее следуют M строк, задающих перегоны. Каждая из этих строк содержит по два целых числа a и b ($1 \leqslant a, b \leqslant N$, $a \neq b$) — номера двух станций, соединённых одним перегоном. Гарантируется, что любая пара (a, b) встретится в этом списке не более одного раза и что если в списке есть пара (a, b), то в списке нет пары (b, a). Также гарантируется, что между любыми двумя станциями можно проехать, используя один или несколько перегонов.

Формат выходных данных

Выведите одно целое число — максимальное количество станций, которые можно посетить, оплатив ровно одну поездку.

Примеры

стандартный ввод	стандартный вывод
5 5 1 2 2 1 1 1 2 1 3 1 4 1 5 2 4	5
5 4 1 2 2 2 1 1 2 1 3 1 4 1 5	4

Замечание

Во втором примере можно, например, сесть на второй станции, через первую доехать до пятой, бесплатно перейти к поезду в обратном направлении, через первую станцию доехать до третьей. Там бесплатно сменить направление не удастся. Максимальное количество посещенных по одному билету станций — 4.

Задача F. Естествоиспытатель

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

Это — интерактивная задача

Естествоиспытатель Вилингстон находится в неизвестной точке с целыми координатами на плоскости, расположенной строго внутри круглого сооружения целого радиуса R. Он может повернуться на какой-то целый угол (от 0 до 359 градусов включительно) и узнать расстояние до границы круга в этом направлении. После каждого поворота Вилингстон возвращается в исходное положение (то есть углы поворотов **не** суммируются.

Требуется помочь ему определить радиус не более, чем за 3 запроса.

Протокол взаимодействия

Взаимодействие начинает программа участника.

Чтобы узнать расстояние до границы круга после поворота на n градусов, выведите `?` n, где n — целое число от 0 до 359 включительно. Угол каждый раз отсчитывается от горизонтального направления, повороты не суммируются. В ответ программа жюри выводит вещественное число с 14 знаками после десятичной точки — расстояние до границы круга в данном направлении. Гарантируется, что радиус круга является целым положительным числом, не превосходящим 10^4, и что точка, из которой делаются запросы, имеет целые координаты и находится строго внутри круга.

Также гарантируется, что значение радиуса не меняется в процессе взаимодействия (то есть что интерактор не является адаптивным).

Чтобы вывести найденный радиус, выведите `!` r, где целое число r является предполагаемым значением радиуса R. Вывод радиуса запросом не считается.

Пример

стандартный ввод	стандартный вывод
5.20714203432655 5.06600306290044 13.53939201416946	? 30 ? 40 ? 270 ! 10

Замечание

В примере запросы производятся из точки с координатами $(3, 4)$. Радиус окружности равен 10.

Не забывайте после каждого запроса или вывода ответа выводить символ перевода строки, а также сбрасывать буфер ввода-вывода. Функция `print` в Python или вывод `endl` в C++ делают это автоматически. Можно также вызывать функцию `flush` используемого языка программирования.

В противном случае ваше решение может получить ошибку «Idleness Limit Exceed».

Задача G. Ё++

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

В компиляторе экспериментального языка отечественной разработки Ё++ реализованы десять различных 64-битных типов данных для хранения вещественных чисел. Тип определяется параметром k ($45 \leqslant k \leqslant 54$). Одно число всегда занимает 64 бита (8 байт), биты занумерованы от самого младшего (0) к самому старшему (63), то есть самый правый бит соответствует значению $2^0 = 1$, а самый левый — 2^{63}.

Значение переменной вычисляется следующим образом:

- Бит 63 является знаковым: если он равен 0, то число неотрицательное, если 1 — отрицательное.
- Биты с 0 по $k-1$ включительно задают мантиссу числа — целое число m ($0 \leqslant m \leqslant 2^k - 1$).
- Биты с k по 62 включительно задают порядок числа — целое число p ($0 \leqslant p \leqslant 2^{63-k} - 1$).
- Если $p = 2^{63-k} - 1$, то содержимое не является обычным числом и обозначается как NaN (Not A Number).
- Если $p > 0$, то значение переменной по абсолютной величине равно $2^{p+1-2^{62-k}} \cdot (1 + m/2^k)$.
- Если $p = 0$, то значение переменной по абсолютной величине равно $2^{1-2^{62-k}} \cdot m/2^k$.

Нумерация бит в числах p и k также идёт от младшего к старшему.

На вход подаётся значение k, определяющее параметр типа, и целое число x — битовое представление переменной. Выведите значение переменной. Если оно равно 0 или целому **нечётному** числу, выведите значение как одно целое число. Если оно равно NaN, выведите текст `NaN`. В оставшихся случаях преобразуйте значение переменной к виду $p \cdot 2^q$, где p — целое **нечётное** число и q — целое число. Если $q < 0$, выведите значение в виде p`/2**`$|q|$, если q положительно — в виде p`*2**`q.

Например, значение -6 будет выведено как `-3*2**1`, 2023 как одно число 2023, $22/16$ как `11/2**3`.

Формат входных данных

Первая строка входных данных содержит одно целое число k ($45 \leqslant k \leqslant 54$) — параметр типа. Вторая строка задаёт содержимое переменной и содержит 64-битное беззнаковое целое число d ($0 \leqslant d \leqslant 2^{64} - 1$).

Формат выходных данных

Выведите значение переменной с заданным содержимым как вещественного 64-битного типа с параметром k.

Примеры

стандартный ввод	стандартный вывод
52 4608871268660281344	11/2**3
52 4656612063538315264	2023
52 13841813454723219456	-3*2**1
46 0	0
54 18446744073709551615	NaN

Задача H. Железнодорожная задача

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

Это — интерактивная задача

Места в плацкартных вагонах пронумерованы следующим образом: нечётные места являются нижними, чётные — верхними, места с номерами с 1 по 36 расположены в основной секции вагона, места с 37 по 54 являются боковыми. Точная нумерация мест приведена на следующей иллюстрации:

Программа жюри загадала некоторое место p.

Ваша задача — отгадать загаданное место с помощью запросов вида `?` d, где d — целое число от -54 до 54 включительно, которое надо прибавить к номеру текущего места. В начале взаимодействия номер текущего места совпадает с загаданным (то есть равен p).

Если номер места, полученный в результате сложения, оказывается меньше 1 или больше 54, то вы **сразу же** проигрываете. В противном случае программа жюри изменяет текущее место, прибавляя к нему d, и выдаёт информацию про это место: является ли оно верхним или нижним, а также является ли оно обычным или боковым.

Требуется не более чем за 6 запросов отгадать **исходное** место p.

Протокол взаимодействия

Взаимодействие начинает ваша программа, отправляя первый запрос.

Каждый запрос имеет формат `?` d, где $-54 \leqslant d \leqslant 54$ — число, которое вы хотите прибавить к номеру текущего места.

Если место некорректно, программа тут же завершается и решение получает вердикт Wrong Answer. В противном случае программа выводит строку из двух слов, разделённых пробелом. Первое слово — `low`, если место нижнее, и `high`, если место верхнее. Второе слово — `main`, если место расположено в основной секции вагона, и `side`, если место боковое.

Чтобы вывести ответ, используйте формат `!` P, где P — предполагаемое значение p. В случае, если ответ будет правильным ($P = p$), и число запросов не превышает 6, решение будет зачтено. Вывод ответа запросом не считается.

Гарантируется, что значение p определено в начале взаимодействия и не меняется в его процессе (то есть интерактор не является адаптивным).

Пример

стандартный ввод	стандартный вывод

`? -1`
 	
`? 5`
 	
`! 33`
	
	`high main`
	
	`low side`

Замечание

В примере из условия программа участника решила вывести ответ наугад, не имея абсолютной уверенности в значении p. В этом случае программе участника повезло, но всегда везти не будет...

Не забывайте после каждого запроса или вывода ответа выводить символ перевода строки, а также сбрасывать буфер ввода-вывода. Функция `print` в Python или вывод `endl` в C++ делают это автоматически. Можно также вызывать функцию `flush` используемого языка программирования.

Задача I. Запись числа

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

Дано целое число, десятичная запись которого не содержит нулей. Требуется найти количество способов разбить десятичную запись этого числа на подстроки ненулевой длины, которые обладают следующим свойством: **не более одного** из чисел, задаваемых этими подстроками, **не** делится на три.

Так как ответ может быть очень большим, выведите остаток от его деления на 998 244 353.

Формат входных данных

Входные данные содержат одно целое положительное число N, десятичная запись которого не содержит нулей ($1 \leqslant N < 10^{100\,000}$).

Формат выходных данных

Выведите одно целое число — остаток от деления количества способов разбить десятичную запись N на блоки, все из которых, кроме, быть может, одного, делятся на три, на 998 244 353.

Примеры

стандартный ввод	стандартный вывод
142857	2
239	4

Задача J. Изделия с орнаментом

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

Во время раскопок в античных Помпеях были найдены изделия с орнаментом, составленным из букв I, V, X, C, L, D, M, обладающим следующим свойством: любые две подряд идущие буквы образуют корректное число в римской записи, но никакие три подряд идущие буквы не образуют корректного числа в римской записи.

Археологи заинтересовались, сколько существует таких орнаментов, состоящих ровно из N букв.

Таблица из Википедии показывает способ корректной записи римских чисел:

Значение разряда	Тысячи	Сотни	Десятки	Единицы
1	M	C	X	I
2	MM	CC	XX	II
3	MMM	CCC	XXX	III
4		CD	XL	IV
5		D	L	V
6		DC	LX	VI
7		DCC	LXX	VII
8		DCCC	LXXX	VIII
9		CM	XC	IX

Заметим, что:

- Числа 4, 9, 40, 90, 400 and 900 записываются в реверсивной нотации, где первый символ вычитается из второго (например, для 40 (`XL`) `X` (10) вычитается из `L` (50)). Это **единственные** места, где реверсивная нотация используется.
- Число, содержащее несколько десятичных цифр, строится дописыванием римского эквивалента каждой цифры от старшего разряда к младшему.
- Если в десятичном разряде стоит 0, никаких цифр в этом разряде в римском представлении не пишется.
- Наибольшее число, которое может быть представлено в римской системе счисления — это число 3,999 (MMMCMXCIX).

Так как ответ может быть очень большим, выведите остаток от его деления на 998 244 353.

Формат входных данных

Первая строка входных данных содержит одно целое число N ($2 \leqslant N \leqslant 3 \cdot 10^6$).

Формат выходных данных

Выведите одно целое число — остаток от деления количества орнаментов длины N на 998 244 353.

Примеры

стандартный ввод	стандартный вывод
2	31
3	28

Задача K. Йода перехватывает информацию

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

В далёкой-далёкой галактике ордену джедаев удалось перехватить секретный канал, который ситхи использовали для обмена сообщениями.

Каждое сообщение состояло из 20 000 строк длины 9, состоящих из цифр от 0 до 9. Магистр Йода сумел выяснить, что приказы, вызывающие эти сообщения, бывают двух типов: «усилить защиту какого-то сектора» и «усилить атаку в каком-то секторе». Кроме того, через своих агентов он узнал следующее.

Сектор задаётся парой целых чисел вида (a, b), где $1 \leqslant a \leqslant b \leqslant 9$. Все числа от $a \cdot 10^8$ до $(b + 1) \cdot 10^8 - 1$ — это номера планет, принадлежащих данному сектору.

Базы защитного флота находятся на всех планетах, у которых номера обладают следующим свойством: они **не кратны 3** и их десятичная запись содержит **хотя бы одну** нечётную цифру.

Базы ударного флота находятся на всех планетах, номера которых являются **простыми** числами.

После выхода каждого приказа ситхи **случайным образом** равновероятно выбирают 20 000 планет среди планет заданного сектора, на которых есть соответствующие базы (базы защитного флота в случае усиления защиты или же базы ударного флота в случае усиления нападения). Затем знаки в десятичной записи каждого номера выбранной планеты случайно переставляются произвольным образом (все возможные варианты перестановки равновероятны), после чего формируется итоговое сообщение.

Вам даётся перехваченное джедаями сообщение — 20 000 строк, каждая из которых состоит из 9 цифр. Известно, что сообщение получено описанным выше способом. Ваша задача — расшифровать приказ (то есть выяснить, усиливается защита или атака и на какой именно сектор распространяется приказ).

Формат входных данных

Входные данные содержат 20 000 строк, каждая из которых содержит одно девятизначное целое число без ведущих нулей. Гарантируется, что все входные данные были получены указанным в условии способом.

Формат выходных данных

Выведите три целых числа: первое должно быть равно 0, если усиливается защита, и 1, если усиливается атака. Далее должны быть выведены два целых числа $a \leqslant b$, задающие сектор.

Примеры

стандартный ввод	стандартный вывод
627353756 098260681 565527418 ... 951068257 107367722 642605294	0 5 6
169743713 544949984 974233805 ... 939373762 898391476 451989221	1 3 5

Задача L. Как стать чемпионом?

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

Чемпионат Нептунианской Баскетбольной Ассоциации (НБА) проходит по следующим правилам: в чемпионате участвуют 2^N команд. 2^{N-1} команд играют в Восточной конференции и 2^{N-1} команд играют в Западной конференции. Сначала играется регулярный сезон, по итогам которого команды занимают места p_i с 1 до 2^N.

Таблица регулярного сезона является **общей**, то есть возможно **произвольное** распределение команд конференций по местам — любое из сочетаний по 2^{N-1} из 2^N. Например, возможен как вариант, когда команды разных конференций в таблице идут через одну — при более-менее равных конференциях, так и вариант, когда в таблице идут сначала все команды одной конференции, а затем — все команды другой (если все команды одной из конференций оказались в кризисе).

Затем играется плей-офф в N кругов. Плей-офф играется по конференциям. Обе конференции до финала обрабатываются независимо. Правила плей-офф в конференциях (первые $N-1$ кругов) определяются следующим образом:

1. Перед началом первого круга команды конференции сортируются по возрастанию номера занятого места, образуя порядок команд $r_{1,*}$ для первого круга при обработке соответствующей последовательности. Последовательности e и w обрабатываются независимо.

2. Каждый круг устроен следующим образом. Берётся порядок $r_{k,*}$ для этого круга. Пусть количество команд в нём равно N_k. Тогда для всех i от 1 до $N_k/2$ команды $r_{k,i}$ и r_{k,N_k+1-i} играют серию между собой, при этом **преимущество домашней площадки** будет у команды, которая в **регулярном** сезоне заняла лучшее место (то есть у которой p_j меньше). Победитель серии выходит в следующий круг с индексом $r_{k+1,i}$.

3. Если после какого-то круга в следующий круг вышла одна команда, она становится победителем конференции и выходит в финал.

Иначе говоря, команда имеет преимущество домашней площадки, если она играет с командой, которая выступила в регулярном чемпионате хуже неё, и не имеет, если играет с командой, которая выступила в регулярном чемпионате лучше неё.

В финале победители Западной и Восточной конференции разыгрывают титул чемпиона, при этом преимущество домашней площадки будет у команды, у которой место p_j в регулярном сезоне меньше.

В аналитической передаче, посвящённой началу сезона, возник вопрос — может ли команда, занявшая место S по итогам регулярного сезона, выиграть чемпионат, имея преимущество домашней площадки ровно в H играх?

Ваша задача — по заданным значениям N, S и H определить, возможна ли такая ситуация, и, если возможна, построить распределение мест в регулярном чемпионате среди команд конференций и результаты всех $2^N - 1$ игр, реализующие эту ситуацию.

Формат входных данных

Первая строка входных данных содержит три целых числа N, S и H ($2 \leqslant N \leqslant 20$, $1 \leqslant S \leqslant 2^N$, $0 \leqslant H \leqslant N$).

Формат выходных данных

В первой строке выведите «`Yes`», если указанный в условии сценарий возможен, и «`No`» в противном случае.

В случае, если ответ равен «`Yes`», выведите схему распределения команд конференций по местам в регулярном сезоне и результаты игр в следующем формате:

В первой строке выведите 2^{N-1} целых чисел от 1 до 2^N — места в регулярном сезоне команд, представляющих Восточную конференцию.

Во второй строке выведите 2^{N-1} целых чисел от 1 до 2^N — места в регулярном сезоне команд, представляющих Западную конференцию. При этом все 2^N чисел в первых двух строках попарно различны.

В третьей строке выведите слово, состоящее из $2^N - 1$ букв. Буква ‘h’ соответствует победе в данной серии команды, которая имеет преимущество домашней площадки (то есть с меньшим местом в регулярном сезоне), буква ‘g’ — победе команды, которая не имеет преимущества домашней площадки (то есть с большим местом в регулярном сезоне).

Первые $2^{N-1} - 1$ букв задают результаты серий Восточной конференции, отсортированных сначала по номеру круга, а потом по индексу, с которым победитель игры выходит в следующий круг, то есть внутри круга k сначала идёт игра, в которой играет команда $r_{k,1}$, затем — в которой играет команда $r_{k,2}$ и так далее. Следующие $2^{N-1} - 1$ букв задают результаты серий Западной конференции в аналогичном формате. Последняя буква задаёт результат финала.

В результате команда, занявшая в регулярном сезоне место S, должна выиграть финал, а количество серий, в которых эта команда имела преимущество домашней площадки, должно быть в точности равно H.

Примеры

стандартный ввод	стандартный вывод
4 16 1	No
4 2 4	Yes 2 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 hhhhghhhghghhgh

Задача M. Летим с комфортом!

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

В международном космопорте «Шариково» $a + b$ мест для космических кораблей. При этом к a местам подведён «посадочный рукав» из терминала, а к b местам пассажирам приходится добираться на автобусе.

Если пассажир добирается из терминала на автобусе, он испытывает одну единицу дискомфорта. Таким образом, пассажиры N-местного корабля, посадка на который проходит с помощью автобуса, испытывают суммарный дискомфорт N.

По действующему регламенту, космический корабль подают под посадку пассажиров на некоторое место в момент времени s_i. Посадка заканчивается к моменту времени $s_i + 1$, затем корабль готовится ко взлету, оставаясь на месте, и взлёт происходит таким образом, что соответствующее место становится свободно с момента времени f_i.

Руководство космопорта решило с целью уменьшения дискомфорта пассажиров изменить регламент.

Теперь после завершения посадки пассажиров корабль может быть однократно перемещён на свободное место до взлёта. При этом:

- Перемещение может произойти в выбранный диспетчером момент t ($s_i + 1 \leqslant t < f_i$).
- Для перемещения нужно, чтобы в момент времени t место, на которое перемещается корабль, было свободно.
- При перемещении место, которое корабль занимал во время посадки пассажиров, освобождается, начиная с момента времени t.
- Из-за перемещения пассажиры N-местного корабля испытывают суммарный дискомфорт $\lfloor N \cdot c \rfloor$, где $0 \leqslant c \leqslant 1$ — некоторая вещественная константа.

По заданной конфигурации космопорта, количеству мест разного типа, количеству кораблей, данных о количестве пассажиров на каждом борту, времени начала посадки и времени старта определите, возможно ли уложиться в расписание, и, если возможно — какой минимальный суммарный дискомфорт испытают пассажиры всех указанных кораблей.

Формат входных данных

Первая строка входных данных содержит одно целое число T — количество сценариев ($1 \leqslant T \leqslant 8$).

В каждом сценарии первая строка содержит три целых числа k, a и b — количество кораблей в расписании, количество мест под посадку, оборудованных рукавом, и количество мест под посадку с доставкой пассажиров автобусом, соответственно ($1 \leqslant k \leqslant 200$, $0 \leqslant a, b \leqslant 30$).

Вторая строка содержит одно целое число c ($0 \leqslant c \leqslant 1$) — коэффициент дискомфорта при перемещении корабля после посадки, заданное не более чем с двумя знаками после десятичной точки. Каждая из последующих k строк содержит по три целых числа N_i, s_i и f_i — количество пассажиров i-го космического корабля, момент начала посадки и момент, к которому корабль взлетит, соответственно ($1 \leqslant N \leqslant 10^5$, $1 \leqslant s_i < f_i - 1 \leqslant 10^9$).

Формат выходных данных

Для каждого сценария выведите одно целое число — минимальный суммарный дискомфорт, который испытают пассажиры всех указанных кораблей.

Если уложиться в расписание невозможно, выведите одно число -1.

Примеры

стандартный ввод	стандартный вывод
2 2 1 0 0.5 20 2 3 1 2 8 6 2 2 0.53 4 2 5 4 3 8 8 5 9 8 5 9 10 6 10 1 8 10	-1 7
2 2 1 1 0.5 10 1 2 20 2 3 2 1 1 0.5 10 1 2 20 1 3	0 10