

Задача А. Успеть на самолёт

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вася сегодня улетаёт на олимпиаду по программированию. Но, к сожалению, у него не очень-то хорошо с тайм-менеджментом, поэтому сейчас он рискует опоздать на свой самолёт. Когда Вася вышел из дома и достал телефон, чтобы включить навигатор, на экране высветилось время: h_1 часов m_1 минут. Навигатор показывает, что до аэропорта Вася доберётся через t_1 минут. Вася помнит, что самолет вылетает в h_2 часов m_2 минут, но в аэропорт нужно приехать заранее, так как регистрация на рейс закрывается за t_2 минут до вылета. Самолёт вылетает в этот же день, и в момент выхода Васи из дома он ещё не улетел. Успеет ли он на самолёт, или нужно уже менять билеты?

Вася — отличный программист, и обязательно бы написал программу, отвечающую на этот вопрос, но он очень торопится в аэропорт в надежде всё-таки успеть. Поэтому вам придётся написать такую программу за него.

Обратите внимание, что, чтобы пройти регистрацию, нужно время. Поэтому, если Вася приедет ровно к её завершению, на рейс он всё равно не попадёт — нужно прийти хотя бы за минуту.

Формат входных данных

В единственной строке входных данных через пробел вводятся числа $h_1, m_1, t_1, h_2, m_2, t_2$ — время в часах и минутах, когда Вася вышел из дома, время на дорогу, время вылета в часах и минутах, а также насколько заранее нужно прибыть в аэропорт на регистрацию ($9 \leq h_1, h_2 \leq 21$, $0 \leq m_1, m_2 \leq 59$, $1 \leq t_1, t_2 \leq 60$). Гарантируется, что самолёт вылетает в тот же день, и в момент выхода Васи из дома самолёт ещё не улетел.

Формат выходных данных

Выведите «OK», если Вася всё же успеет на самолёт, и «CHANGE TICKET», если придётся менять билет.

Примеры

стандартный ввод	стандартный вывод
9 0 10 9 10 10	CHANGE TICKET
10 0 50 15 30 45	OK

Задача В. Квадраты и прямые

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

Дано n квадратов на плоскости, пронумерованных от 1 до n , причем вершины i -го квадрата — точки с координатами $(i, 0)$, $(0, -i)$, $(-i, 0)$, $(0, i)$ (то есть центр каждого квадрата лежит в $(0, 0)$, а его стороны пересекают оси координат под углом 45°).

Дано q прямых. Для каждой прямой необходимо найти суммарное количество точек пересечения этой прямой со всеми квадратами. При этом гарантируется, что никакая прямая не пересекает ось Ox под углом 45° . Заметьте, что если прямая проходит через вершину квадрата, эта вершина считается за точку пересечения.

Формат входных данных

Первая строка входных данных содержит два целых числа n и q ($1 \leq n \leq 10^9$, $1 \leq q \leq 10^5$).

Каждая из следующих q строк содержит описание прямой: четыре целых числа x_1, y_1, x_2, y_2 — координаты двух точек, лежащих на прямой ($-10^9 \leq x_1, y_1, x_2, y_2 \leq 10^9$). Гарантируется, что эти две точки не совпадают (то есть $x_1 \neq x_2$ или $y_1 \neq y_2$). Также гарантируется, что прямая, проведенная через эти две точки, не пересекает ось Ox под углом 45° .

Формат выходных данных

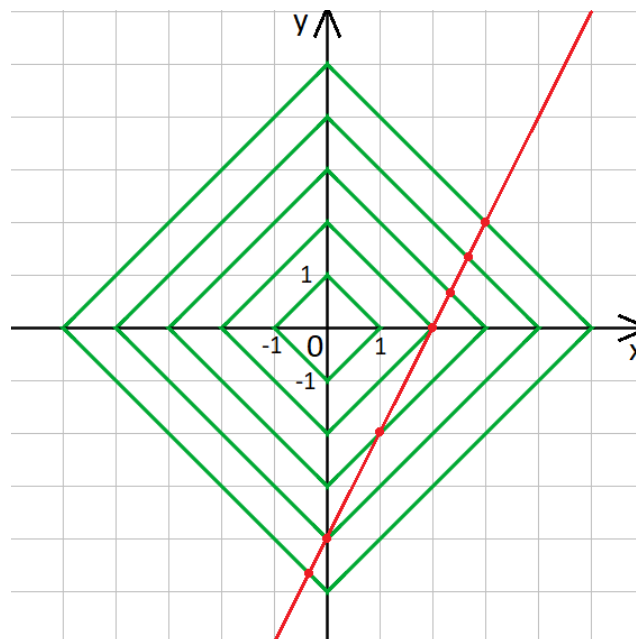
Для каждой прямой выведите в отдельной строке одно целое число — количество точек пересечения прямой с квадратами.

Пример

стандартный ввод	стандартный вывод
5 4	7
4 4 1 -2	10
0 0 -3 1	10
-1 0 1 0	0
7 0 0 6	

Пояснение к примеру

Для первого примера и первой прямой рисунок выглядит так:



Задача С. Разрезание на квадраты

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

Разрежьте квадрат на n квадратов — возможно, разного размера — или определите, что это невозможно.

Формат входных данных

В первой строке задано целое число n — количество квадратов в разрезании ($1 \leq n \leq 50$).

Формат выходных данных

Если разрезать квадрат на n квадратов невозможно, выведите «NO». Иначе в первой строке выведите «YES», а в следующих n строках — квадраты разрезания в любом порядке.

Чтобы задать разрезание, следует расположить квадраты так, что их стороны параллельны координатным осям, а координаты вершин целые. Каждый квадрат задаётся тройкой целых чисел « $x \ y \ s$ » — координатами левого нижнего угла и длиной стороны ($0 \leq x, y, s \leq 10^9$, $s > 0$). Например, «4 2 5» задаёт квадрат с вершинами в точках (4, 2), (4, 7), (9, 2) и (9, 7). Можно показать, что, если разрезание квадрата на $n \leq 50$ квадратов вообще существует, то существует и разрезание с такими ограничениями.

Выведенный набор квадратов должен являться разрезанием квадрата: у каждой пары квадратов в наборе не должно быть общих внутренних точек, а объединение всех квадратов набора должно также быть квадратом.

Примеры

стандартный ввод	стандартный вывод
2	NO
4	YES 0 0 100000000 100000000 100000000 100000000 100000000 0 100000000 0 100000000 100000000

Задача D. Умножение на три

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

Сегодня Таня решила попрактиковаться в умножении на 3.

У Тани есть любимая цифра x , и девочка написала эту цифру на доске. Потом Таня умножила записанную цифру на 3 и написала последнюю цифру результата справа от первой. Затем она снова умножила только что записанную цифру на 3, написала последнюю цифру результата справа от первых двух. Таня продолжала так до тех пор, пока на доске не оказалось записано n цифр.

И теперь Таню интересует вопрос: будет ли полученное длинное число делиться на 3?

Формат входных данных

Единственная строка ввода содержит два целых числа x и n — любимую цифру Тани и количество цифр в итоговом числе ($1 \leq x \leq 9$, $1 \leq n \leq 10^9$).

Формат выходных данных

Выведите «YES», если полученное число делится на 3, или «NO», если не делится.

Пример

стандартный ввод	стандартный вывод
2 3	NO

Пояснение к примеру

В первом примере Таня сначала записала цифру 2, затем $2 \cdot 3 = 6$ и затем 8 ($6 \cdot 3 = 18$). У нее получилось число 268, которое не делится на 3.

Задача Е. Игра в деление

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

На доске записано целое число $n > 1$. Ивонна и Зара играют в игру, делая ходы по очереди, начинает Ивонна. За каждый ход игрок может либо поделить число на 2 и округлить вниз, либо поделить на 3 и округлить вверх. После этого старое число стирается, а новое записывается на доску.

Выигрывает та, кто первой запишет на доске число 1. Кто выиграет при правильной игре?

Формат входных данных

В первой строке задано целое число n ($1 < n \leq 10^{18}$).

Формат выходных данных

Если при правильной игре выигрывает Ивонна, выведите «Yvonne».

Если при правильной игре выигрывает Зара, выведите «Zara».

Буквы можно выводить как в верхнем, так и в нижнем регистре.

Примеры

стандартный ввод	стандартный вывод
2	Yvonne
4	Zara

Задача F. Количество делителей

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дан массив a из n целых чисел. Поступают q запросов двух типов:

1. Изменить значение a_i на x ;
2. Посчитать количество чисел среди $a_l, a_{l+1}, \dots, a_r$, которые являются делителями числа x .

Формат входных данных

Первая строка содержит два целых числа n и q — длина массива и число запросов соответственно ($1 \leq n, q \leq 10^5$).

Вторая строка содержит целые числа $a_1, a_2, \dots, a_n$ — начальные значения чисел в массиве ($1 \leq a_i \leq 10^5$).

Каждая из следующих q строк задаёт запрос. Запрос первого типа записывается в формате «1 i x » ($1 \leq i \leq n$; $1 \leq x \leq 10^5$), а второго — в формате «2 l r x » ($1 \leq l \leq r \leq n$; $1 \leq x \leq 10^5$).

Формат выходных данных

Выведите ответ на каждый запрос второго типа. Гарантируется, что есть хотя бы один запрос второго типа.

Пример

стандартный ввод	стандартный вывод
5 7	4
1 2 3 4 5	1
2 1 5 12	4
2 2 3 4	2
1 3 5	0
2 1 5 10	
1 3 7	
2 2 4 20	
2 5 5 11	

Задача G. Планы на вечер

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Аня очень любит дорешивать задачи олимпиадного кружка по программированию. Сегодня ей осталось решить n задач. Решение i -й задачи отнимает у Ани a_i сил, но после успешной сдачи Аня воодушевляется и получает b_i сил. Кроме задач, у Ани есть школьное домашнее задание по m предметам. Домашнее задание по j -му предмету отнимает c_j сил для того, чтобы его сделать.

Изначально у Ани есть t сил. Аня может решать задачи и делать школьные уроки в любом порядке, но она не начинает решать задачу или делать домашнее задание по какому-то предмету, если это отнимет больше сил, чем у неё есть. Начав решать задачу или делать домашнее задание, Аня доводит дело до конца. Времени у Ани достаточно, чтобы решить все задачи и сделать все домашние задания.

Может ли Аня действовать так, чтобы за вечер решить все задачи и сделать все домашние задания? Если нет, то может ли она хотя бы сделать все домашние задания?

Формат входных данных

Первая строка содержит три целых числа n , m , t — количество задач по программированию, количество школьных уроков и изначальное количество сил у Ани ($1 \leq n, m \leq 15$, $1 \leq t \leq 1000$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ — силы, которые Аня тратит на задачи по программированию ($1 \leq a_i \leq 50$).

Третья строка содержит n целых чисел $b_1, b_2, \dots, b_n$ — силы, которые Аня получает после решённой задачи ($1 \leq b_i \leq 50$).

Четвёртая строка содержит m целых чисел $c_1, c_2, \dots, c_m$ — силы, которые Аня тратит на школьные предметы ($1 \leq c_j \leq 50$).

Формат выходных данных

Выведите:

- «SUPER», если Аня может спланировать свой вечер таким образом, чтобы сдать все задачи и сделать все уроки;
- «SCHOOL», если Аня может спланировать время так, чтобы хотя бы сделать все уроки;
- «EPIC FAIL», если не существует плана, при котором Аня сделает уроки.

Примеры

стандартный ввод	стандартный вывод
4 2 3 6 4 5 3 2 7 4 5 2 1	SUPER
2 3 4 9 6 10 12 2 2 1	EPIC FAIL
2 3 4 4 3 2 6 3 2 2	SCHOOL

Пояснения к примерам

В первом примере количество сил у Ани равно 3, и она может делать задания в такой последовательности:

- решить задачу 4, после чего у неё будет сил $3 - 3 + 5 = 5$ сил;
- решить задачу 3, тогда у неё будет $5 - 5 + 4 = 4$ сил;
- решить задачу 2, тогда у неё будет $4 - 4 + 7 = 7$ сил;
- сделать домашнее задание 2, у неё останется $7 - 1 = 6$ сил;
- решить задачу 1, у неё останется $6 - 6 + 2 = 2$ сил;
- сделать домашнее задание 1, на которое у неё как раз остались силы.

Задача Н. Ножницы

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

Маше на день рождения подарили красивые и острые ножницы для разрезания прямоугольников. Теперь она очень хочет опробовать их в деле! Для этого у неё есть прямоугольник $m \times n$, который она хочет разрезать на прямоугольники размера $1 \times k$. Удастся ли ей задуманное?

Формат входных данных

Единственная строка входных данных содержит три целых числа m , n и k ($1 \leq m, n, k \leq 10^9$).

Формат выходных данных

Выведите «YES», если Маше удастся разрезать прямоугольник, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
2 3 2	YES
2 3 3	YES
6 6 4	NO

Задача I. Рафтинг

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

В горах Берляндии есть n прекрасных озёр с чистой прозрачной водой и m бурных горных рек, вытекающих из одних озёр и впадающих в другие. Благодаря редким горным породам и сложным механизмам, по которым на протяжении многих миллионов лет формировались русла рек и бассейны озёр, горные водоёмы имеют совершенно уникальную структуру. Для любой пары озёр в горах Берляндии существует не более одной реки, текущей из первого во второе. Кроме того, озёра расположены на разной высоте относительно уровня мирового океана, поэтому для любых двух озёр, если из первого озера во второе течёт река, то не существует пути из второго озера в первое, который бы следовал по течению рек. Так как горные озёра небольшие, для любого озера существует не более двух рек, истекающих из него, и не более одной реки, впадающей в него.

Эти живописные места особенно ценят поэты и любители рафтинга — спортивного сплава по рекам на надувных судах (рафтах).

Аня и Боря очень любят рафтинг, поэтому сегодня они отправились в горы Берляндии со своим рафтом, чтобы сплавиться по некоторым из рек. Они очень не любят что-либо планировать заранее, поэтому для начала своего маршрута они выбирают случайное озеро, в которое не впадает ни одна река. Кроме того, когда им в процессе встречается развилка, реку, по которой ребята будут двигаться дальше, они также выбирают случайным образом.

Вася — поэт и куда меньший любитель активного отдыха, чем Аня и Боря, поэтому он хочет просто сидеть на камне на берегу, наслаждаться природой и ждать своих друзей у того из озёр, в котором они закончат сплав, — то есть у озера, из которого не вытекает ни одна река. Вася понимает, что может быть много озёр, в которых ребята могут завершить сплав, поэтому он хочет ждать своих друзей около того озера, в котором это произойдёт с наибольшей вероятностью. Помогите Васе выбрать такое озеро.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество озёр и количество рек ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$). Озёра пронумерованы целыми числами от 1 до n .

В каждой из следующих m строк записано по два целых числа u и v , обозначающих наличие реки из озера u в озеро v ($1 \leq u, v \leq n$, $u \neq v$).

Гарантируется, что, если из озера u в озеро v течёт река, то не существует пути из v в u , который бы следовал по течению рек. Также гарантируется, что из любого озера истекает не более двух рек и в любое озеро впадает не более одной реки.

Формат выходных данных

Выведите одно целое число — номер озера, у которого следует сидеть Васе. Если подходящих номеров озёр несколько, выведите наименьший из них.

Пример

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

Задача J. Иван и дороги

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

Иван живёт в центре Санкт-Петербурга, и ему часто приходится ездить по городу для решения важных дел.

Чтобы разгрузить центр города, губернатор постановил, что все дороги будут односторонними, а за езду в неправильном направлении придётся заплатить штраф, зависящий от конкретной дороги. Но иногда бывает невозможно добраться от одного места до другого, не заплатив при этом ни одного штрафа.

Иван очень бережно относится к своим финансам, поэтому просит вас вычислить, какое минимальное количество рублей ему придётся отложить на свою следующую поездку.

Более формально: центр Санкт-Петербурга представляется в виде n перекрёстков и m односторонних дорог. Для каждой дороги предусмотрена величина штрафа за проезд в неправильном направлении. Выведите, сколько Ивану придётся отложить рублей на поездку от перекрёстка A до перекрёстка B , или -1 , если от A до B доехать невозможно.

Формат входных данных

В первой строке содержится четыре целых числа n, m, A, B — количество перекрёстков, дорог, начало и конец пути соответственно ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$, $1 \leq A, B \leq n$).

Затем идут m строк. В каждой записаны по три целых числа u, v, c , задающие одностороннюю дорогу от перекрёстка u к перекрёстку v со штрафом c , если ехать по ней в обратном направлении ($1 \leq u, v \leq n$, $1 \leq c \leq 10^9$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 5 1 3 1 2 3 3 2 7 4 1 2 5 4 1 3 5 3	6
3 3 1 3 1 2 5 2 3 10 3 1 2	0
2 0 2 1	-1

Пояснения к примерам

В первом примере Иван может проехать по перекрёсткам 1-4-5-3 и заплатить $2 + 1 + 3 = 6$ рублей.

Задача К. Речной бой

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Саша и Женя играют в «Речной бой». В этой игре у каждого игрока есть поле — полоска из 20 клеток — которое держится в секрете от другого игрока. Поле Саши белое, а поле Жени чёрное.

В начале игры игрок ставит на своё поле четыре корабля: по одному кораблю из одной, двух, трёх и четырёх клеток. Клетки каждого корабля должны следовать на поле подряд, а клетки различных кораблей не могут быть соседними. Вот пример начальной расстановки кораблей: 00022033300000444401. Здесь цифры 1–4 соответствуют клеткам кораблей такой длины, а нули — пустые клетки.

Далее игроки делают ходы по очереди. Начинает Саша. В свой ход игрок выбирает несколько клеток на поле противника, чтобы одновременно атаковать эти клетки. Игрок может выбрать столько клеток, сколько их в его самом длинном не потопленном корабле, или любое меньшее их количество. Корабль потоплен, если все его клетки уже атакованы. Например, на первом ходу Саша может выбрать четыре клетки для атаки: ????!???!?????????!. Здесь вопросительные знаки — клетки на поле Жени, про которые Саша ничего не знает, а восклицательные знаки — клетки, выбранные для атаки.

После выбора противник должен рассказать, что было в каждой атакованной клетке — они могли быть пусты или содержать части кораблей. Например, Женя могла ответить на атаку Саши так: ????0???44?????????1. Это означает, что в левой атакованной клетке было пусто, в двух средних были части корабля длины 4, а в самой правой — единственная часть корабля длины 1. Теперь Женин корабль длины 1 потоплен, так как все его клетки атакованы, а корабль длины 4 ещё не потоплен, ведь атакованы только две из его четырёх клеток. Следующей ходит Женя, и она может также выбрать четыре клетки для атаки.

Выигрывает тот, кто первым потопит все четыре корабля противника. После этого победитель показывает своё поле противнику, чтобы тот мог убедиться, что корабли на нём расставлены по правилам.

Саша — на самом деле робот. В начале игры он выбирает начальное положение случайно и равномерно из всех возможных. На каждом ходу Саша выбирает клетки для атаки одну за другой случайно и равномерно из всех не атакованных клеток, пока не кончатся либо такие клетки, либо возможности его собственных кораблей. Например, если у Саши остался не потопленным только корабль из двух клеток, то он может выбрать только две клетки для атаки.

Женя — на самом деле тоже робот, но она программируемая. Напишите программу для Жени, чтобы победить Сашу 50 раз! В этой задаче 50 тестов, в каждом из них нужно одержать победу. Поведение Саши в каждом тесте выбрано случайно, как написано выше, но зафиксировано заранее и одинаково для всех проверок на этом тесте.

Протокол взаимодействия

В игре программа участника и программа жюри обмениваются строками, в которых записано положение на поле. Каждый ход состоит из двух строк — вопроса и ответа.

Изначально каждый игрок ничего не знает про поле другого. Поле в таком случае записывается как «*color* ??????????????????», где *color* — это «*black*» для чёрного поля (Саши) и «*white*» для белого поля (Жени). Клетки, выбранные для атаки, будут обозначаться восклицательными знаками («!»). После атаки клетки, которые оказались пустыми, обозначаются нулями («0»), а клетки, в которых находятся части кораблей — цифрой, равной размеру корабля («1–4»). При следующих ходах клетки, атакованные ранее, всё так же содержат цифры, а не атакованные всё так же содержат вопросительные знаки — даже если цифру на клетке уже можно восстановить однозначно. Поле всегда состоит ровно из 20 клеток.

В ход Саши сначала прочитайте строку вида «*black* ...» — чёрное поле, как о нём известно Сахе, с выбранными клетками для атаки (их будет не больше, чем в самом большом не потопленном

корабле Саши). Затем выведите строку вида `«black ...»` — то же чёрное поле, но во всех выбранных для атаки клетках отметьте, частью какого корабля они являются.

В ход Жени сначала выведите строку вида `«white ...»` — белое поле, как о нём известно Жене, с выбранными клетками для атаки (их должно быть не больше, чем в самом большом не потопленном корабле Жени). Затем прочитайте строку вида `«white ...»` — то же белое поле, но во всех выбранных для атаки клетках будет отмечено, частью какого корабля они являются.

Если все корабли Жени потоплены, Саша выводит строку вида `«white ...»` — расстановку кораблей на белом поле без вопросительных и восклицательных знаков. Гарантируется, что Саша выбрал её в начале игры, и каждая строка, выведенная программой жюри, согласована с этой расстановкой. После этого проверка на тесте завершается с вердиктом `«Wrong answer»`.

Если все корабли Саши потоплены, Женя должна вывести строку вида `«black ...»` — расстановку кораблей на чёрном поле без вопросительных и восклицательных знаков. Каждая строка, выведенная решением, должна быть согласована с этой расстановкой. Если это так, проверка на тесте завершается с вердиктом `«OK»`, если нет — с вердиктом `«Wrong answer»`. Как можно заметить, Саша никак не проверяет, что Женя выбрала расстановку в самом начале игры, а не ставила корабли во время ответов.

После вывода каждой строки, чтобы не получить вердикт `«Idleness limit exceeded»`, следует вывести перевод строки и очистить буфер вывода. Последнее можно сделать, например, вызовом `fflush (stdout)` в C или C++, `sys.stdout.flush ()` в Python, `flush (output)` в Pascal или `System.out.flush ()` в Java.

Пример

стандартный ввод	стандартный вывод
black ?!!????????????!?! white ???04???0???1????? black ?00??!?!?????!!0??0 white ?3?004??40????1????0 black ?00!!0?3?!???000?!0 white ?330044440????1?2??0 black ?00000!3?0?!!?000?20 white 3330044440?00?1?22?0	black ?00???????????0??0 white ????!?!?!?!?!?!? black ?00??0?3?????000??0 white ?!?!04??!0????1????! black ?00000?3?0???000?20 white ?3!004!!40????1?!?0 black ?0000033?0?44?000?20 white !330044440?!!?1?2!?0 black 10000033304444000220

Задача L. Outro

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

Даны два массива строк: массив a длиной n и массив b длиной m . Вам нужно выбрать **непустую** строку, которая является подстрокой всех строк массива a и является подстрокой минимального количества строк массива b . Причем из всех таких строк вам нужно найти строку минимальной длины.

Вам нужно вывести лишь два числа: сначала количество строк из массива b , для которых выбранная вами строка является подстрокой, а затем длину выбранной строки (вам нужно сначала минимизировать первое число, а затем второе). Если выбрать такую строку невозможно, выведите $-1 -1$.

Формат входных данных

Первая строка содержит одно целое число T — количество тестовых случаев ($1 \leq T \leq 10$).

Каждый тестовый случай начинается со строки, которая содержит одно целое число n — количество строк в массиве a ($1 \leq n \leq 50\,000$).

В следующих n строках записаны элементы массива a , по одному на строке. Каждая строка в массиве a имеет длину от 1 до 500 символов включительно и состоит только из строчных букв английского алфавита.

Далее следует строка, содержащая одно целое число m — количество строк в массиве b ($1 \leq m \leq 50\,000$).

В следующих m строках записаны элементы массива b , по одному на строке. Каждая строка в массиве b имеет длину от 1 до 500 символов включительно и состоит только из строчных букв английского алфавита.

Гарантируется, что сумма длин всех строк в массиве a по всем тестовым случаям не превосходит 50 000. Также гарантируется, что сумма длин всех строк в массиве b по всем тестовым случаям не превосходит 50 000.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите по два числа — либо ответ на задачу, либо $-1 -1$, если ответа не существует.

Пример

стандартный ввод	стандартный вывод
3	1 1
3	0 7
abc	0 1
cab	
aba	
3	
abb	
acc	
aaa	
1	
abacaba	
2	
abacab	
bacaba	
1	
alice	
1	
bob	

Задача М. Покупка дивана

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

Вася собирается устроить вечеринку у себя дома и пригласить n друзей. Но, к сожалению, у него нет мебели, на которой все могли бы удобно разместиться. В магазине поблизости от Васиного дома продаются диваны, длина каждого из которых равна l сантиметров. Каждый из друзей Васи занимает на диване x сантиметров. Когда несколько человек сидят рядом, между ними должно быть расстояние хотя бы y сантиметров.

Какое минимальное количество диванов надо купить Васе, чтобы все его друзья уместились? Для самого Васи у него дома есть удобное кресло.

Формат входных данных

Единственная строка содержит четыре целых числа: n — количество друзей Васи, x — число сантиметров, которое занимает на диване каждый из друзей, y — минимальное расстояние между двумя людьми на диване, l — длина дивана ($1 \leq n \leq 100$, $1 \leq x, y \leq 200$, $x \leq l \leq 1000$).

Формат выходных данных

Выведите одно число — минимальное количество диванов, которое надо купить Васе.

Пример

стандартный ввод	стандартный вывод
5 200 100 500	3

Задача N. Подготовка к Новому году

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

Алиса и Боб с нетерпением ждут новогодних праздников, и уже сейчас начали к ним готовиться. Сегодня они пришли в магазин за ёлкой и игрушками. Игрушки они уже выбрали, но, как оказалось, для одной ёлки их слишком много. Поэтому дети решили купить сразу три ёлки. Конечно, чтобы получилось красиво, нужно выбрать ёлки одинаковой высоты.

В магазине есть n ёлок, пронумерованных целыми числами от 1 до n . Помогите Алисе и Бобу найти подходящий набор ёлок или скажите, что такого не существует.

Формат входных данных

Первая строка входных данных содержит одно целое число n — количество ёлок в магазине ($1 \leq n \leq 100$).

Вторая строка содержит n целых чисел $h_1, h_2, \dots, h_n$ — высоты ёлок ($1 \leq h_i \leq 100$).

Формат выходных данных

В случае, если подходящего набора не существует, просто выведите -1 в единственной строке.

В ином случае выведите три различных целых числа в любом порядке — номера трёх ёлок, которые следует выбрать. Если существует несколько подходящих наборов, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
6 1 2 3 2 1 2	2 4 6
5 1 2 3 4 5	-1

Задача О. Крестики

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

Миша вырезал полоску бумаги из n клеточек, а затем в некоторых клеточках поставил крестики так, чтобы никакие два крестика не оказались рядом. После этого он отдал полоску бумаги своему брату Лёше и попросил его дорисовать крестиков, но при этом никакие два крестика по-прежнему не могли стоять рядом. Затем они отдали полоску их сестре Кате, и она сказала, что больше крестиков по их правилам поставить нельзя.

Известно, где поставил крестики Миша. Какое максимальное количество крестиков мог поставить Лёша?

Формат входных данных

Первая строка содержит целое число n — длину полоски ($1 \leq n \leq 1000$).

Вторая строка задаёт полоску, которая получилась у Миши. Она состоит из n символов: i -й символ равен «X» (большая английская буква «икс»), если в i -й клеточке бумаги нарисован крестик, или «.» (точка), если эта клетка пуста. Гарантируется, что никакие два крестика не стоят рядом.

Формат выходных данных

Выведите одно число — максимальное количество крестиков, которое мог поставить Лёша.

Примеры

стандартный ввод	стандартный вывод
6	3
9 ...X...X	2
17X..X.....X..	5