

Problem A. Checkers game

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Для проведения тренировок по шашкам компания «Train Sporting» разработала пакет специальных программ. Одна из таких программ проверяет, все ли возможные ходы видит игрок. Вам поручено написать основной модуль, находящего все ходы, возможные в некоторой позиции на доске.

В пакете программ используются следующие правила игры в шашки:

1. Игра проходит на доске 8×8 клеток. Вертикали пронумерованы буквами латинского алфавита от a до h , горизонтали — цифрами от 1 до 8. Доска раскрашена в черный и белый цвета таким образом, что клетка $a1$ является черной, и любые две соседние (по вертикали или горизонтали) клетки имеют разный цвет.
2. В начале игры у каждого игрока есть по 12 шашек соответствующего цвета (черного или белого). Белые шашки занимают все черные клетки на горизонталях 1, 2 и 3, черные соответственно занимают все черные клетки на горизонталях 6, 7 и 8.
3. Ход игрока заключается в том, чтобы произвести ход шашкой, взятие, либо последовательность взятий одной из своих шашек.
4. Ход шашкой состоит в перемещении ее на соседнюю по диагонали черную клетку (несложно видеть, что в игре будут использоваться исключительно черные клетки доски). При этом перемещать шашку на занятую своей или чужой шашкой клетку нельзя. При ходе белой шашкой номер горизонтали должен увеличиваться, а для черной — уменьшаться.
5. Взятие шашкой состоит в перемещении ее по диагонали на две клетки в любую сторону. При этом на той клетке, через которую проходит шашка, должна находиться шашка противоположного цвета, а клетка, на которую шашка попадает, должна быть свободной.
6. Последовательность взятий шашкой представляет собой любое ненулевое количество взятий, выполняемых одно за другим. При этом запрещается брать шашку, уже взятую в этой последовательности.
7. После выполнения хода все взятые шашки убираются с доски.
8. Игрок, который не может сделать ход, проигрывает.
9. Как только белая шашка достигает восьмой горизонтали (черная, соответственно, первой), она становится дамкой. После этого такая шашка может ходить (и бить) не только с увеличением (уменьшением) номера горизонтали, но и в обратную сторону. Также дамке разрешается при ходе (и взятии) перемещаться не на одну (две клетки), а на произвольное количество клеток. При этом все клетки, через которые проходит дамка, должны быть свободны. При взятии, все клетки, через которые проходит дамка, кроме последней, должны быть свободны (равно как и клетка, на которую дамка попадет в результате хода). На предпоследней клетке, через которую проходит дамка, должна быть шашка противоположного цвета.
10. Во всех случаях взятия дамки берутся по тем же правилам, что и обычные шашки. Разрешается в течение последовательности взятий брать шашки и дамки в любом порядке.

Вам задана позиция — некая расстановка шашек на доске. Выведите все ходы белых, возможные в этой позиции.

Input

Во входном файле задано восемь строк по восемь символов в каждой — позиция, которую требуется проанализировать. Каждый символ соответствует одной клетке доски. Если клетка пуста, ей соответствует символ '.', если в ней находится белая пешка, то символ 'w', черная пешка — 'b', белая дамка — 'W', черная дамка — 'B'. Строки входного файла соответствуют горизонталям в порядке от 8 до 1, символы в строках — вертикалям в порядке от *a* до *h*.

Output

Выведите все возможные ходы белых. Ходы требуется выводить в соответствии со стандартной нотацией: ход без взятия отмечается координатами клеток, из которой переместилась пешка (дамка) и в которую переместилась пешка (дамка), выведенных через дефис, например *a1-b2*. Последовательность взятий обозначается координатами начальной клетки, а также всех клеток, на которые пешка (дамка) попадает после очередного взятия, выведенных через двоеточие, например *a1:c3:e1*.

Ходы выводите в произвольном порядке. Следуйте как можно точнее формату примера.

Example

standard input	standard output
<pre>b.b. ...w.... ...B... w..... </pre>	<pre> d4:f6 d4:f6:h4 d4:f2 d4-c5 a1-b2 </pre>
<pre>b..b.W.... ...b.b. .b..... </pre>	<pre> d4:g7 d4:a1 d4:f2 d4:f2:h4 d4:f2:h4:e7 d4:f2:h4:e7:a3 d4:f2:h4:e7:a3:c1 d4-c5 d4-b6 d4-a7 d4-e5 d4-c3 </pre>

Problem B. Memory training

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Для тренировки памяти была предложена следующая методика. На длинной бумажной ленте записана последовательность (a_i) , состоящая из l_a чисел. Лента разрезается на n частей таким образом, чтобы ни один разрез не проходил через запись какого-нибудь числа и чтобы каждая часть содержала не менее m и не более k чисел. Для каждой части определим её индекс как номер (начиная с 1) элемента последовательности (a_i) , перед которым был произведён разрез. Задание состоит в том, чтобы по названной последовательности (s_i) определить, существует ли какая-то часть, на которой записана в точности эта последовательность, и, если существует, то попытаться назвать её индекс.

Ваша задача несколько проще. По заданным последовательностям (a_i) и (s_i) определите, возможна ли при заданных n , m и k ситуация, когда на какой-то части будет записана только последовательность s , и ничего, кроме неё?

Input

В первой строке входного файла задано целое положительное число $T \leq 1000$ — количество тестовых примеров. В первой строке тестового примера задано число l_a ($1 \leq l_a \leq 10^5$) — длина исходной последовательности. Во второй строке заданы l_a чисел a_i , ($-10^9 \leq a_i \leq 10^9$) — элементы исходной последовательности. В третьей строке заданы три числа m , k и n ($1 \leq n \leq l_a$, $1 \leq m \leq k \leq l_a$, $m \times n \leq l_a \leq k \times n$) — соответственно ограничения снизу и сверху на длины подпоследовательностей, записанных на частях, полученных при разрезании ленты, содержащей последовательность a , и количество таких частей. В четвёртой строке задано одно число l_s ($m \leq l_s \leq k$) — длина последовательности (s_i) . В пятой и последней строке тестового примера заданы l_s чисел s_i ($-10^9 \leq s_i \leq 10^9$) — элементы последовательности (s_i) . Гарантируется, что размер входного файла не превышает 5 мегабайт.

Output

Для каждого тестового примера в выходной файл выведите в отдельной строке **NO**, если ни при одном разрезании ленты с последовательностью (a_i) , удовлетворяющем условию задачи, нельзя на одной из частей получить только последовательность (s_i) . Иначе выведите **YES** и минимальный индекс той части, на которой записана последовательность (s_i) .

Example

standard input	standard output
1 6 8 9 10 11 12 13 1 4 3 2 9 10	YES 2
2 6 9 10 11 12 13 14 2 3 3 2 10 11 6 8 2 2 2 4 3 2 3 3 2 2 2	NO YES 3

Problem C. 3-2-1

Input file: **standard input**
Output file: **standard output**
Time limit: 8 seconds
Memory limit: 64 Mebibytes

Перед Чемпионатом Мира по футболу ФИФА с целью повышения эффективности работы судей провела тренировку по следующим правилам. Судья находится в углу прямоугольного поля размера $m \times n$, где m и n — целые неотрицательные числа. Судья стоит в точке $(0, 0)$, стороны поля параллельны осям координат. Во всех остальных целочисленных точках внутри и на сторонах поля стоят футболисты. Судья может достать красную карточку и показать в каком-то направлении. При этом все футболисты, на которых указывает рука судьи, держащая карточку, уходят с поля. За какое минимальное число показов карточек судья удалит всех футболистов?

При вычислениях считать, что судья и футболист являются точками, поворот руки судьи задаёт направление. Рука судьи указывает на некоторого футболиста, если прямая, идущая в заданном направлении и проходящая через точку $(0, 0)$, проходит через точку, соответствующую этому футболисту.

Input

В первой строке входного файла задано целое положительное число $T \leq 10$ — количество тестовых примеров. В последующих T строках перечислены тестовые примеры. Каждый тестовый пример содержит два целых числа m и n ($1 \leq m, n \leq 10^6$) — размеры поля.

Output

Для каждого тестового примера выведите в отдельной строке, какое минимальное число показов карточек потребуется судье для того, чтобы удалить всех игроков.

Example

standard input	standard output
1 3 2	7
1 5 4	17
3 4 5 7 8 9 11	17 41 70

Problem D. Sort and swap

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 Mebibytes

Во время тренировки перед районной олимпиадой по информатике на доске одним из школьников был написан следующий пример функции, которая сортирует массив:

```
void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

void sort (int m[], int L)
{
    int i, j;
    for (i = 0; i < L; ++i)
        for (j = 0; j < L; ++j)
            if (m[i] > m[j]) swap(&m[i], &m[j]);
}
```

По окончании тренировки школьников этот фрагмент остался на доске. Когда на очередную тренировку пришли студенты, им было предложено выяснить, сколько раз при работе функции будет вызвана процедура `swap` при работе с выписанным ниже на доске массивом.

Напишите программу, которая по заданному массиву вычисляет количество вызовов функции `swap` при сортировке этого массива.

Input

В первой строке входного файла задано число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке задан сам массив — n чисел a_i , ($-10^9 \leq a_i \leq 10^9$);

Output

В выходной файл выведите одно целое число — количество вызовов процедуры `swap` при сортировке заданного массива.

Examples

standard input	standard output
6 5 7 9 11 13 15	15
6 15 13 11 9 7 5	10

Problem E. Stones

Input file: **standard input**
Output file: **standard output**
Time limit: 2 секунды
Memory limit: 64 Mebibytes

Для тренировки сотрудников фирм, занимающихся продажей компьютерных комплектующих, была придумана следующая игра. Игра начинается с того, что в ряд выкладываются n процессоров Intel и AMD. На каждом ходу игрок может сделать одно из следующих действий:

1. Убрать k «камней» из начала ряда.
2. Убрать k «камней» из конца ряда.
3. Убрать каждый второй «камень», начиная с самого левого.
4. Убрать каждый второй «камень», начиная со второго слева.

Каждое из действий разрешается сделать только в том случае, если в результате его выполнения остается не меньше «камней», чем было убрано на данном шаге. Игра заканчивается, как только остается всего один «камень». Если производителем этого «камня» является Intel, то выиграл первый игрок, если AMD — второй.

Вам требуется определить, кто из игроков выиграет, если известно, что они действуют оптимальным образом.

Input

В первой и единственной строке входного файла содержится n ($1 \leq n \leq 400$) символов 'A' и 'I' — исходный расклад процессоров, где 'A' обозначает процессор AMD, а 'I' — процессор Intel.

Output

В первой строке выходного файла выведите 1, если выиграет первый игрок, иначе выведите 2.

Examples

standard input	standard output
AIII	1
AIAI	1
AIAIA	2

Problem F. STAGE

Input file: **standard input**
Output file: **standard output**
Time limit: **3 seconds**
Memory limit: **64 Mebibytes**

В некоторой системе машинного перевода языковые конструкции представлены в виде деревьев разбора. Исходная фраза разбивается на несколько семантических блоков, каждый из которых, в свою очередь, может состоять из нескольких семантических блоков и так далее. Каждому семантическому блоку сопоставляется его тип — одна из строчных латинских букв a, b, c, d, e .

Будем считать, что дерево разбора A вложено в дерево разбора M , если существует такое соответствие $f(x)$ между вершинами дерева A и некоторыми вершинами дерева M , что:

- каждой вершине x дерева A соответствует ровно одна вершина $f(x)$ дерева M , а каждой вершине дерева M соответствует не более одной вершины дерева A ;
- вершина x принадлежит поддереву дерева A с корнем в вершине y тогда и только тогда, когда вершина $f(x)$ принадлежит поддереву дерева M с корнем в вершине $f(y)$;
- для всех вершин x тип вершины x совпадает с типом вершины $f(x)$.

Для «тренировки» системы, в частности, требуется реализовать следующую подзадачу: задано дерево разбора M и несколько запросов. Один запрос состоит в том, чтобы для некоторого дерева разбора A_i выяснить, вложено ли дерево A_i в дерево M . Требуется обработать все запросы в порядке поступления

Input

В первой строке входного файла задано дерево разбора M , состоящее из не менее, чем одной, и не более, чем из 10^4 вершин.

Дерево задаётся следующим способом: для каждого семантического блока, начиная с основной фразы, задан его тип — одна из строчных латинских букв a, b, c, d, e , затем, если блок разбивается на несколько семантических блоков — задан список таких блоков. Список заключён в круглые скобки, а описания блоков в списке разделены запятой. Так, описание $e(e, d(b, b), a)$ обозначает, что основная фраза имеет тип e и разбивается на три блока — первый из них имеет тип e и не содержит других семантических блоков, второй имеет тип d и разбивается на два семантических блока, каждый из которых имеет тип b и не содержит других семантических блоков, третий имеет тип a и не содержит других семантических блоков.

Во второй строке входного файла указано количество запросов n ($1 \leq n \leq 100$).

В каждой из последующих n строк входного файла задан один запрос — дерево разбора A_i , содержащее не менее одной и не более 16 вершин.

Output

Для i -го запроса в отдельной строке выведите 1, если дерево A_i вложено в дерево M , и 0 в противном случае.

Example

standard input	standard output
$e(e, d(b, b), a)$	1
2	0
$e(b, a)$	
$e(d, b)$	