

## Задача А. Слоники в магазине

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

В магазине слоников — сезонная распродажа!

Слоники выстроены на витрине в один ряд.

Первый покупатель купил  $a$  слоников, стоящих правее всех; второй покупатель купил  $b$  слоников, стоящих левее всех; третий покупатель купил  $c$  самых высоких из оставшихся слоников.

При каком минимальном исходном числе слоников в магазине возможная такая ситуация?

### Формат входных данных

Первая строка входных данных содержит одно целое число  $a$ . Вторая строка входных данных содержит одно целое число  $b$ . Третья строка входных данных содержит одно целое число  $c$  ( $1 \leq a, b, c \leq 10^4$ ).

### Формат выходных данных

Выведите одно целое число — минимальное исходное число слоников в магазине, при котором возможна описанная ситуация.

### Пример

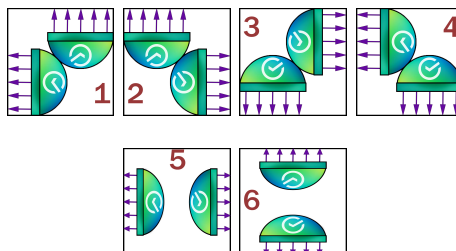
стандартный ввод	стандартный вывод
21 11 2021	2053

## Задача В. Склады СберМаркет

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

Поддержка работы крупного онлайн-магазина, доставляющего товары по всей стране, такого, как СберМаркет, требует организации огромных складов. Склады расположены под землёй и обслуживаются автоматическими тележками. На некоторых участках тележки должны подниматься на поверхность. Для этого устанавливаются специальные впуск-выпускные ворота.

На участке  $1 \times 1$  километр допустимы следующие конфигурации установки ворот:



Для СберМаркета оборудуется новый центральный склад. Территория склада представляет собой огороженный прямоугольник  $w$  на  $h$  километров. В каждом секторе размера  $1 \times 1$  километр **нужно** разместить одну из 6 конфигураций ворот.

Из стороны, где располагаются ворота, проведены фиолетовые стрелочки.

Автоматические тележки двигаются строго по прямой от одних ворот до других; в случае встречи двух тележек, едущих навстречу, они разъезжаются с помощью системы предотвращения столкновений и продолжают двигаться по той же прямой каждая в прежнем направлении.

Автоматические тележки могут выезжать из любых ворот и въезжать в любые ворота; также в конфигурациях 5 и 6 тележка может проехать между воротами по вертикали для конфигурации 5 и по горизонтали для конфигурации 6.

Если тележка попадает в ворота с обратной стороны или врежется в ограждение склада, происходит авария.

Требуется указать для каждого сектора склада, какая из 6 конфигураций ворот должна в нём располагаться, чтобы аварии были невозможны.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $w$  и  $h$  ( $1 \leq w, h \leq 10^3$ ) — ширина и высота склада СберМаркета.

### Формат выходных данных

Выведите  $h$  строк, в каждой строке  $w$  чисел в диапазоне от 1 до 6, разделённые пробелами — номера конфигураций ворот в соответствующих квадратах склада. Если ответов несколько, выведите любой.

Если расположить ворота без риска аварий невозможно, выведите  $-1$ .

### Примеры

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

## Задача С. Сериал и реклама

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

В небольшом американском городке снимается сериал «Cops And Donuts».

В меню местной пекарни, в которой происходит действие очередного эпизода, есть  $n$  сортов пончиков. Пончики готовятся по заказу, то есть в любой момент времени любой покупатель может выбрать любой пончик.

По сюжету, в течение дня в магазин планирует зайти  $n$  копов и купить ровно по одному пончику. Владелец пекарни уже планирует рекламную кампанию и считает сорт популярным, если пончик этого сорта куплен хотя бы одним копом. На данный момент  $k$  копов купили пончики.

У вас есть данные о том, какой коп купил какой пончик. Определите наибольшее и наименьшее количество сортов пончиков, которые в итоге могут стать популярными.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество копов (оно же — количество пончиков).

Вторая строка входных данных содержит одно целое число  $k$  — количество копов, которые уже купили пончики ( $1 \leq k \leq n$ ).

Каждая из последующих  $k$  строк содержит по одному целому числу между 1 и  $n$  включительно — тип пончика, купленный очередным копом.

### Формат выходных данных

В первой строке выведите одно целое число — наименьшее количество сортов пончиков, которые могут стать популярными после того, как все  $n$  копов купят по пончику, во второй строке выведите одно целое число — наибольшее количество сортов пончиков, которые могут стать популярными после того, как все  $n$  копов купят по пончику.

### Пример

стандартный ввод	стандартный вывод
25	3
4	24
20	
21	
11	
21	

## Задача D. Яндекс.Панорамы космоса

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Для того, чтобы построить карты на Яндекс-панорамах, используются специальные автомобили, оборудованные фотоаппаратурой, которые ездят по дорогам страны и фотографируют окружающие пейзажи.

Но в рамках нового проекта Яндекс.Космос планируется добавить на Яндекс.Панорамы фотографии из дальнего космоса. Автомобили в космос отправлять уже не оригинально, так что для этого был разработан специальный космический аппарат «Яндексспутник». Аппарат пролетает через пояс астероидов и делает снимки по следующему алгоритму.

1. Изначально количество снимков равно 0;
2. Окрестности пояса астероидов представляются как такое подпространство трёхмерного декартового пространства, что координаты  $x, y, z$  принимают значение от 0 до  $10^9$ .
3. Космический аппарат летит по орбите, задаваемой уравнением  $(ax+by+cz)^2+cx+dy+ez+1 = n$
4. Точки съёмки расположены в точках орбиты, обладающим следующим свойством: все координаты являются целыми неотрицательными числами, а их произведение равно фиксированному целому числу  $L$ .
5. Фотография, сделанная в точке съёмки с координатами  $x, y$  и  $z$ , передаётся на Землю за  $x$  секунд (на время этих  $x$  секунд включается мощный передатчик).

Требуется вычислить суммарное время работы мощного передатчика за время картографирования пояса астероидов.

### Формат входных данных

В первой строке заданы 5 целочисленных значений – соответствующие координаты  $a, b, c, d, e$  ( $1 \leq a, b, c, d, e \leq 30$ ).

В следующей строке заданы значения  $n$  и  $L$  ( $1 \leq n \leq 10^9, 0 \leq L \leq 10^{18}$ ).

Гарантируется, что минимальное расстояние между любыми двумя соседними точками съёмки в любой из проекций не превосходит 200.

### Формат выходных данных

Выведите одно целое число — суммарное время работы передатчика.

### Примеры

стандартный ввод	стандартный вывод
3 1 10 8 1 1000 0	1
3 1 10 8 1 1000000000 40019525280	1440

### Замечание

Во втором примере одна из точек съёмки — (1440, 1059, 26243).

## Задача Е. Малотупоугольность

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

Какое **наименьшее** количество тупых углов может быть у строго выпуклого  $n$ -угольника?  
 $n$ -угольник называется строго выпуклым, если он является выпуклым и никакие три вершины не лежат на одной прямой.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $3 \leq n \leq 10^6$  — количество углов рассматриваемого многоугольника.

### Формат выходных данных

Выведите одно целое число — ответ к задаче.

### Пример

стандартный ввод	стандартный вывод
3	0

## Задача F. Дерево бонсай

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

Всеволод – бывший олимпиадник, раньше на соревнованиях он испытывал широкий спектр эмоций: волнения, азарт, радость от победы, но сейчас все это в прошлом. Сейчас, в воскресный дождливый день, он сидит на своем широком диване напротив камина и читает книгу. Его взгляд падает на ветвленный пожилой бонсай...

Бонсай Всеволода – подвешенное двоичное дерево, каждая вершина которого является листом или имеет *ровно* два потомка. Корень дерева имеет номер 1, а все оставшиеся вершины пронумерованы числами от 2 до  $N$ . В каждом листе изначально записано число 1. *Интересностью* вершины назовем наибольший общий делитель чисел, записанных в листах, лежащих в поддереве заданной вершины. Требуется отвечать на запросы следующих видов:

1. Умножить число, записанное в вершине  $v$  на  $x$ , гарантируется, что  $v$  – лист;
2. Вывести интересность вершины  $v$  по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке входных данных содержатся числа  $N$  и  $Q$  ( $1 \leq N, Q \leq 3 \cdot 10^5$ ) – размер дерева и количество запросов, соответственно.

В следующей строке записано  $N - 1$  число  $p_i$  – номера предков вершин с номерами  $2, 3, \dots, N$  соответственно ( $1 \leq p_i \leq N$ ).

Далее в  $M$  строках описаны запросы, отвечать на которые требуется последовательно, каждый запрос относится к одному из двух типов. «1  $v$   $x$ » значит, что число в листе с номером  $v$  умножить на  $x$  ( $1 \leq v \leq N, 1 \leq x \leq 10^6$ ). А запрос вида «2  $v$ » означает, что надо вывести интересность вершины с номером  $v$  по модулю  $10^9 + 7$  ( $1 \leq v \leq N$ ).

### Формат выходных данных

Для каждого запроса типа 2 в отдельной строке выведите интересность соответствующей вершины по модулю  $10^9 + 7$ .

### Пример

стандартный ввод	стандартный вывод
3 10	1
1 1	13
1 2 13	13
1 3 44340	2
1 3 22553	26
2 1	13
2 2	
2 3	
1 2 2	
2 1	
2 2	
2 3	

## Задача G. Спидран по Рас-Ман

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

Недавно Федя стал победителем всероссийской олимпиады, проходившей в офисе 1С в Москве, и ему выплатили большую премию. Он не знал, куда ее потратить, поэтому решил купить редкие устройства и пытаться пройти на них новейшие компьютерные игры. И вот недавно 1С gaming выпустила игру под названием «Сибирский РасМан» для устройств, совместимых с Siemens SL45. Потратив половину премии на покупку раритетного устройства в рабочем состоянии, Федя поставил игру и приступил к делу. Дело в том, что Федя – *спидраннер*, то есть он занимается скоростным прохождением видеоигр. Федя, конечно же, и сам мог придумать оптимальную стратегию для прохождения этой игры, но он занят просмотром «Милого во Франксе».

В «Сибирском Рас-Ман» игрок управляет Рас-Ман – главным героем, который изначально находится в левой верхней клетке прямоугольного поля размером  $r \times c$  клеток, в каждой из которых находится по одной точке. Рас-Ман должен съесть все точки, после чего прийти в клетку, лежащую в  $a$ -й строке и  $b$ -м столбце, на этом игра и закончится. Если хотя бы раз за игру Рас-Ман побывает в какой-то клетке, то он автоматически съест точку, в ней находящуюся. Требуется вывести строку, состоящую из символов 'U', 'L', 'D', 'R', обозначающую последовательность ходов, которые должен совершить Рас-Ман для того, чтобы пройти игру. **Эта строка должна иметь минимально возможную длину.**

Символ 'U', обозначает, что Рас-Ман должен переместиться в клетку, которая находится над клеткой, в которой он сейчас находится, 'D' обозначает переход вниз, 'L' – влево, 'R' – вправо. Строки нумеруются сверху-вниз, а столбцы – слева-направо. Нумерация строк и столбцов начинается с единицы, так что стартовая клетка имеет номер строки и столбца, равный 1.

### Формат входных данных

В первой строке задано единственное число  $t$  ( $1 \leq t \leq 600$ ) – количество наборов входных данных.

В каждой из последующих  $t$  строк задан один тестовый пример, состоящий из четырёх целых чисел  $r$  и  $c$  ( $2 \leq r, c \leq 5000$ ), а потом  $a$  и  $b$  ( $1 \leq a \leq r$ ,  $1 \leq b \leq c$ ,  $a \cdot b \neq 1$ ).

Так же гарантируется, что сумма значений  $r \cdot c$  по всем входным данным не превосходит  $3 \cdot 10^6$ .

### Формат выходных данных

Для каждого набора данных в отдельной строке выведите строку, состоящую из символов 'L', 'R', 'D', 'U', записанную без пробелов – ответ на задачу. Если вариантов ответа несколько, выведите любой.

### Примеры

стандартный ввод	стандартный вывод
2 3 3 2 2 2 2 1 2	RRDDLLUR DRU
1 2 3 1 3	DRRULR

## Задача Н. Самый умный

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

В неизвестном лице «три семерки» учатся  $N$  учеников, каждый из которых изначально считает себя самым умным. Ученики занумерованы последовательными целыми числами от 1 до  $N$ .

Потом происходят  $M$  уроков, по одному уроку каждый день. Каждый урок описывается тройкой чисел  $(p, x, y)$ , где  $1 \leq p, x, y \leq N$ .

На урок приходит учительница и спрашивает у ученика с номером  $p$ , кого он считает самым умным, тот ей честно отвечает, что самый умный ученик имеет номер  $q$ . Далее, если  $p = q$ , то ученика вызывают к доске, а иначе учительница спрашивает уже у ученика с номером  $q$ , кто самый умный и процесс продолжается в том же ключе. В конце концов учительница либо кого-нибудь вызывает к доске, либо после того, как задаст вопрос некоторому ученику второй раз, устраивает скандал. После этого урока ученик с номером  $x$  кардинально меняет свое мнение и впоследствии считает самым умным ученика с номером  $y$ . Для каждого числа  $i$  от 1 до  $M$  выведите номер ученика, которого вызовут к доске в  $i$ -й день, а также количество вопросов, которые при этом задаст учительница, или  $-1$ , если учительница устроит скандал.

### Формат входных данных

В первой строке входного файла содержатся числа  $N$  и  $M$  ( $1 \leq N, M \leq 3 \cdot 10^5$ ). Каждая из последующих  $M$  строк содержит по три целых числа  $p, x$  и  $y$  ( $1 \leq p, x, y \leq N$ ).

### Формат выходных данных

В последующих  $M$  строках требуется вывести результаты опросов учеников для каждого дня. Если учительница в  $i$ -й день устроит скандал, выведите в  $i$ -й строке число  $-1$ . Иначе, выведите пару чисел – номер ученика, которого вызвали к доске в  $i$ -й день и количество заданных вопросов соответственно.

### Пример

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



## Задача I. Хорошие раскраски – 3

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

Это интерактивная задача.

Маленький Ильдар очень любит раскраски. В этот замечательный день к нему в гости приехал Кирилл. Кирилл недавно выиграл олимпиаду и в подарок ему предоставили премию. К сожалению, премия была не очень большой, но Кирилл все же очень хотел порадовать маленького Ильдара, поэтому купил ему  $t$  тетрадных листов  $100 \times 100$  клеток и два карандаша красного и черного цвета. Ильдар принял подарок и раскрасил все свои листы в два цвета, но Кириллу рисунки решил не показывать, а сыграть с ним в игру.

Кириллу предлагается найти четыре одноцветные клетки, центры которых образуют прямоугольник, стороны которого параллельны линиям сетки. Он может спрашивать у Ильдара, в какой цвет он покрасил какую-нибудь клетку, но, чтобы играть было интереснее, ему разрешается задать не более, чем 15 вопросов. Они играют  $t$  партий, причем каждый раз на новом листе.

### Протокол взаимодействия

В первой строке будет записано одно число  $t$  ( $t \leq 2000$ ) – количество игр, которые хотят провести Ильдар и Кирилл, все игры **независимы** друг от друга и будут проводиться на разных листах.

Чтобы задать вопрос про очередную клетку, следует вывести  $? x y$  в отдельной строке, где  $x$  и  $y$  – номер строки и столбца, где лежит данная клетка ( $1 \leq x, y \leq 100$ ). На каждый ваш вопрос Ильдар будет отвечать символом  $B$  или  $R$ , в зависимости от того, в какой цвет покрасил соответствующую клетку ( $B$  – черный,  $R$  – красный). Если число вопросов превысило 15 или клетка лежит за границами листа, Ильдар ответит FAIL.

Когда вы будете готовы назвать четыре одноцветные клетки, выведите  $! x_1 y_1 x_2 y_2$  в отдельной строке, сообщая, что клетки  $(x_1, y_1)$ ,  $(x_1, y_2)$ ,  $(x_2, y_1)$ ,  $(x_2, y_2)$  покрашены в одинаковый цвет (разумеется,  $1 \leq x_1, y_1, x_2, y_2 \leq 100$ ,  $x_1 \neq x_2$ ,  $y_1 \neq y_2$ ). После этого Ильдар ответит Вам OK или FAIL в зависимости от того, угадали вы или нет. Если он ответил OK, то автоматически вы начинаете играть следующую партию.

В случае если Ильдар вам выведет FAIL после какого либо вашего запроса (первого или второго типа), Вам следует немедленно завершить работу вашей программы. В противном случае вместо вердикта WA Вы можете получить другие значения ошибки (TL, RE, IL).

Обратите внимание, что в данной задаче интерактор является адаптивным, то есть состояние раскраски всегда консистентно уже сделанным запросам, но в остальном может меняться в процессе работы.

### Пример

стандартный ввод	стандартный вывод
1	
R	? 1 1
B	? 1 2
R	? 1 3
R	? 3 1
R	? 3 3
OK	! 1 1 2 2

## Замечание

Для корректной работы программы после каждой операции вывода запроса или вывода ответа требуется выводить символ переноса строки, а также очищать буфер вывода, то есть делать следующие операции:

- В языке Pascal: **flush(output);**
- В C/C++: **fflush(stdout)** или **cout.flush();**
- В Java: **System.out.flush();**
- В Python: **sys.stdout.flush()** из библиотеки **sys**;
- В C#: **Console.Out.Flush();**

## Задача J. Лестница деда Василия

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

За долгое время жизни в деревне у деда Василия накопилось большое количество поленьев. Они стоят в один ряд во дворе, для каждой из них хозяйственный дед измерил высоту и пронумеровал их от 1 до  $n$ .

Теперь ему захотелось часть из них пустить в дело, а именно построить лестницу, такую, чтобы она начиналась с полена высотой 1 и далее каждая последующая была выше предыдущей на 1. Дед решил, что нужно выбрать несколько поленьев соответствующей высоты, извлечь их из ряда и соорудить лестницу.



Так как поленьев во дворе много, то Василий не хочет сильно нарушать их исходный порядок и будет брать поленья таким образом, что после постройки лестницы в исходном ряду будет не более одного пустого места, то есть все выбранные поленья для будущей лестницы должны образовывать непрерывный отрезок от  $l$ -й до  $r$ -й.

Теперь Василий не может заснуть, так как задумался: а насколько высокую лестницу он сможет сделать при таких условиях?

### Формат входных данных

В первой строке находится одно целое число  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ) — количество поленьев во дворе деда Василия.

Во второй строке содержится  $n$  целых чисел  $a_i$  через пробел, где  $a_i$  — высота  $i$ -го полена в ряду ( $1 \leq a_i \leq n$ ).

### Формат выходных данных

Вывести одно число — высоту наибольшей возможной лестницы, которую можно построить при условии, что все выбранные для её постройки поленья образуют непрерывный отрезок в исходном ряду.

### Пример

стандартный ввод	стандартный вывод
19 9 5 2 1 3 4 7 2 5 6 5 8 1 3 7 2 4 10 1	8

### Замечание

В примере из условия можно взять поленья из отрезка от 2 до 6, получится лестница высоты 5. Можно взять поленья из отрезка от 4 до 10, получится лестница высоты 7. Но самая большая лестница получится, если взять отрезок от 10 до 17. Тогда высота лестницы будет равна 8.

## Задача К. Бутерброды

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

Раньше Саша был олимпиадником, но из-за того, что он живет не в столице, а в небольшом городе К, он не смог на этом заработать денег, и теперь ему приходится работать поваром в студенческой столовой. Сейчас Саша занят составлением рецептов бутербродов, которые в будущем он планирует подавать на завтрак. Изначально перед ним лежит  $K$  кусков хлеба и он совершает с ними  $N$  последовательных операций, каждая из которых относится к одному из следующих типов:

1. Саша берет ингредиент  $x$  из холодильника и кладет сверху на  $i$ -й бутерброд;
2. Саша снимает с  $i$ -го бутерброда верхний ингредиент и возвращает его в холодильник;
3. Саше начинает казаться, что  $i$ -й и  $j$ -й бутерброды слишком сильно отличаются по размеру, поэтому он начинает перекладывать ингредиенты с большего бутерброда на меньший, пока разница их размеров станет не больше, чем 1.

Требуется написать программу, которая выведет количество различных бутербродов, которые получались у Саши в ходе составления рецептов. Пустой бутерброд не считается бутербродом. Бутерброды, которые получаются в процессе выполнения операции третьего типа, также следует принимать во внимание.

Два бутерброда считаются различными, если у них различается число ингредиентов или их порядок (то есть существует такое  $i$  что на  $i$ -м сверху месте расположены ингредиенты разного типа).

### Формат входных данных

В первой строке указаны два числа  $K$  и  $N$  ( $K \leq 20$ ,  $N \leq 3 \cdot 10^5$ ) – количество кусков хлеба, лежащих перед Сашей и количество действий, соответственно.

В последующих  $N$  строках описаны действия, совершенные Сашей. Каждая строка начинается с числа 1, 2 или 3, обозначающих тип операции. Если операция имеет тип 1, то после написаны два целых числа  $i$  и  $x$  ( $1 \leq i \leq K$ ,  $1 \leq x \leq 10^9$ ) – номер бутерброда и ингредиента, соответственно. Для операции типа 2 записано одно целое число  $i$  ( $1 \leq i \leq K$ ) – номер бутерброда. Гарантируется, что на этом бутерброде есть хотя бы один ингредиент. Наконец, для операций типа 3 записана пара целых чисел  $i$  и  $j$  ( $1 \leq i, j \leq K$ ,  $i \neq j$ ) – номера бутербродов, которые надо «уравнять».

### Формат выходных данных

Вывести требуется единственное число – ответ на задачу.

### Пример

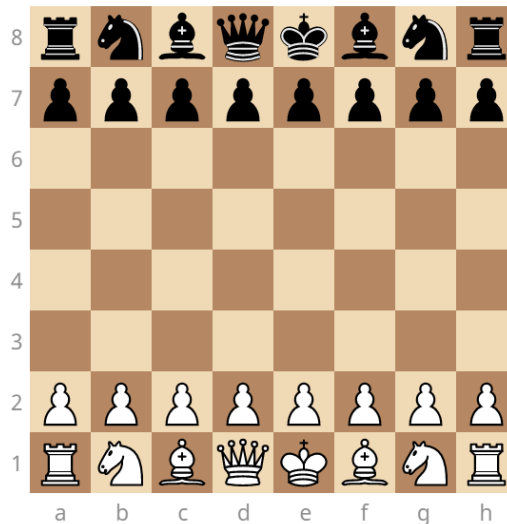
стандартный ввод	стандартный вывод
3 5 1 1 3 1 1 3 3 2 1 1 2 7 1 3 7	4

## Задача L. Дрон над шахматной доской

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

Это интерактивная задача.

Над шахматной доской в начальной позиции летает дрон. Изначально дрон висит над какой-то клеткой и может перемещаться на одну клетку влево, вправо, вверх и вниз. Если дрон пытается выйти за пределы доски, он остаётся на месте (при этом никаких сообщений не получает).



С помощью камеры он может узнать, что находится под ним. Если поле свободно, система сообщает цвет поля. Если поле занято, система сообщает тип фигуры (но не цвет фигуры, ни цвет поля не сообщается). Камера срабатывает каждый раз после попытки перемещения, а также в момент включения дрона над стартовым полем.

Стартовое поле, над которым висит дрон, неизвестно. Ваша задача — не более, чем за 8 ходов определить это поле.

### Формат входных данных

Информация с камеры дрона передаётся Вашей программе в следующем формате.

- Если дрон висит над пустой белой клеткой, Вы получаете сообщение «white».
- Если дрон висит над пустой чёрной клеткой, Вы получаете сообщение «black».
- Если дрон висит над клеткой с любой пешкой, Вы получаете сообщение «pawn».
- Если дрон висит над клеткой с любой ладьёй, Вы получаете сообщение «rook».
- Если дрон висит над клеткой с любым конём, Вы получаете сообщение «knight».
- Если дрон висит над клеткой с любым слоном, Вы получаете сообщение «bishop».
- Если дрон висит над клеткой с любым ферзём, Вы получаете сообщение «queen».
- Если дрон висит над клеткой с любым королём, Вы получаете сообщение «king».

### Формат выходных данных

Чтобы переместить дрон, используйте следующие команды:

- ? U — переместиться вверх по доске;
- ? D — переместиться вниз по доске;
- ? R — переместиться вправо по доске;
- ? L — переместиться влево по доске.

Чтобы вывести ответ, используйте команду `! cell`, где *cell* — обозначение поля в шахматной нотации, то есть **a1** — левое нижнее поле, горизонтали занумерованы от *a* до *h* снизу вверх, вертикали — от 1 до 8 слева направо.

## Протокол взаимодействия

Сначала Ваша программа получает информацию с камеры дрона об исходной клетке.

Затем Вы перемещаете дрон, после каждого перемещения получаете информацию с камеры дрона о той клетке, в которой оказались в результате (в частности, если была попытка уйти с доски, то дрон остаётся на месте и Вы получаете информацию о клетке, с которой делался ход). Перемещать дрон Вы можете не более 8 раз.

Как только Вы готовы вывести ответ, выводите его. Вывод ответа не включается в число ходов.

Взаимодействие не является адаптивным, то есть стартовое поле для каждого теста однозначно определено перед стартом взаимодействия и не меняется.

## Пример

стандартный ввод	стандартный вывод
white	? L  ? L  ? D  ? D  ? D  ! b3
black	
black	
pawn	
rook	
rook	
rook	

## Замечание

Для корректной работы программы после каждой операции вывода запроса или вывода ответа требуется выводить символ переноса строки, а также очищать буфер вывода, то есть делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout)` или `cout.flush();`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush()` из библиотеки `sys`;
- В C#: `Console.Out.Flush();`

## Задача М. Внимание, нули!

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

В двоичной записи целого положительного числа, не содержащей ведущих нулей,  $N$  знаков, из которых ровно  $K$  нулей. Какое максимальное и минимальное число нулей может быть в шестнадцатеричной записи этого числа, не содержащей ведущих нулей?

### Формат входных данных

Первая строка входных данных содержит одно целое число  $N$  ( $1 \leq N \leq 10^5$ ) — количество знаков в исходном двоичном числе. Вторая строка входных данных содержит одно целое число  $K$  — количество нулей в исходном двоичном числе ( $0 \leq K \leq N - 1$ ).

### Формат выходных данных

Сначала выведите максимальное количество нулей в шестнадцатеричной записи  $N$ -значного двоичного числа с ровно  $K$  нулями в десятичной записи, затем выведите минимальное их количество.

### Пример

стандартный ввод	стандартный вывод
21	2 0
11	