

## Задача A. Restore Permutation

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

**Это задача с двойным запуском.** Ваше решение будет запущено два раза.

При первом запуске вам дается  $p$  — перестановка целых чисел от 1 до  $n$ . Ваша программа должна вывести битовую строку (строку из нулей и единиц) длины не более  $m$ . Число  $m$  не известно вашей программе и зависит от подзадачи, соответствующее значение указано в таблице системы оценивания. Если ваша программа выведет строку длиннее, чем  $m$ , она получит вердикт «Wrong Answer».

Между запусками решения программа жюри поменяет местами два различных элемента исходной перестановки, получив перестановку  $q$ .

Во втором запуске на вход вашей программе подается перестановка  $q$  и выведенная вами в первом запуске битовая строка. Требуется восстановить исходную перестановку.

### Формат входных данных

При первом запуске первая строка ввода содержит число 1 и целое число  $n$  — длину перестановки ( $2 \leq n \leq 10^6$ ). Во второй строке через пробел перечислены  $n$  различных целых чисел  $p_i$  — элементы перестановки ( $1 \leq p_i \leq n$ ).

При втором запуске первая и вторая строки в том же формате содержат число 2, длину перестановки  $n$  и перестановку  $q$ , а третья строка содержит строку из нулей (символ '0') и единиц (символ '1') длины не более  $m$ .

### Формат выходных данных

При первом запуске выведите битовую строку длины не более  $m$ , которую затем будете использовать для восстановления перестановки при втором запуске.

При втором запуске выведите через пробел  $n$  различных целых чисел от 1 до  $n$  — элементы исходной перестановки  $p$ .

### Протокол взаимодействия

Во избежание получения некорректных вердиктов вроде `Idleness Limit Exceeded` или `Security Violation` заканчивайте вывод каждой строки символом перевода строки ('`\n`').

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
0	–	примеры из условия, $m = 2000$	
1	7	$n \leq 16, m = 2000$	0
2	15	$n \leq 10^5, m = 2 \cdot 10^6$	
3	13	$n \leq 2000, m = 200$	0, 1
4	11	$m = 32\,000$	2
5	17	$m = 7000$	2, 4
6	10	$m = 2000$	0 – 2, 4, 5
7	19	$m = 500$	0 – 2, 4 – 6
8	18	$m = 200$	0 – 7

## Примеры

стандартный ввод	стандартный вывод	Пояснение
1 4 4 1 2 3	100001010011	Первый запуск
2 4 4 3 2 1 100001010011	4 1 2 3	Второй запуск

стандартный ввод	стандартный вывод	Пояснение
1 5 2 3 4 5 1	111100100111010100011111100110 00100010011011110001111111010 011100100	Первый запуск
2 5 2 3 5 4 1 111100100111010100011111100110 00100010011011110001111111010 011100100	2 3 4 5 1	Второй запуск

## Замечание

В примерах длинные битовые строки выведены с переводом строки для корректного отображения в условии. На самом деле этих переводов строк нет. Также учтите, что ввод и вывод при втором запуске зависят от вывода при первом запуске, и не обязаны совпадать с показанными в условии примерами.

## Задача В. 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 ОК или FAIL в зависимости от того, угадали вы или нет. Если он ответил ОК, 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 С. Советские ясли

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

У Михаила Абрамовича, члена КПСС с 1961 года, доктора наук, научного атеиста, любящего мужа и отца, 10 лет назад родился внук по имени Максим. Но в отличие от своего дедушки Максим не увлекается науками, а вместо этого дни напролет играет в игры на телефоне.

Недавно Максим скачал себе новую игру на телефон — «Змейка 2022». Игровое поле для «Змейки 2022» представляет собой прямоугольную таблицу  $n \times m$ . Клеткой  $(r, c)$  будем называть клетку, которая находится на пересечении  $r$ -й строки и  $c$ -го столбца, пронумеруем строки сверху вниз от 1 до  $n$ , столбцы — слева направо от 1 до  $m$ . В каждой клетке поля находится яблоко, которое змейка съедает, когда ее голова оказывается в этой клетке. При этом игрок получает  $w_{ij}$  очков, когда змейка съедает яблоко, которое находится в клетке  $(i, j)$ .

В начале игры змейка имеет длину 1, ее голова появляется в клетке  $(a_r, a_c)$  и немедленно съедает расположенное там яблоко. Игра заканчивается, когда голова змейки оказывается в клетке  $(b_r, b_c)$ .

За один ход игрок перемещает голову змейки в любую из соседних клеток, которая пока не занята змейкой. Так как в каждой клетке находится яблоко, то змейка наращивает свою длину на 1 после каждого хода и множество клеток, занятых змейкой остается тем же, плюс к нему добавляется клетка, в которую переместилась её голова. Ход змейки описывается одним символом: «U» для перемещения вверх, из  $(r, c)$  в  $(r - 1, c)$ ; «D» для перемещения вниз, из  $(r, c)$  в  $(r + 1, c)$ ; «L» для перемещения влево, из  $(r, c)$  в  $(r, c - 1)$ ; «R» для перемещения вправо, из  $(r, c)$  в  $(r, c + 1)$ .

Пусть  $W$  — суммарная стоимость яблок внутри поля, тогда считается, что игрок победил, если набрал строго больше  $\frac{1}{2}W$  очков. При этом для упрощения игры гарантируется, что любое яблоко приносит строго меньше очков, чем суммарно приносят яблоки в клетках  $(a_r, a_c)$  и  $(b_r, b_c)$ .

Для Максима данная игра оказалась слишком сложной, он никак не может выиграть. Поэтому он обратился за помощью к своему дедушке. Михаил Абрамович же в ответ рассказал историю, как во время его молодости точно такую же задачу решил обычный советский детсадовец.

Вы выступаете в роли этого детсадовца, ваша задача заключается в том, чтобы для каждой конфигурации игрового поля из тестов предьявить выигрышную стратегию.

### Формат входных данных

В первой строке дано число  $t$  — количество тестов во входных данных.

Каждый из тестов описывается в следующем формате. В первой строке указаны шесть чисел  $n, m, a_r, a_c, b_r, b_c$  — размеры поля, координаты начальной клетки и координаты конечной клетки ( $2 \leq n, m \leq 5000, 1 \leq a_r, b_r \leq n, 1 \leq a_c, b_c \leq m$ , начальная и конечная клетки различны).

Сумма  $n \cdot m$  по всем тестам в одном наборе входных данных не превышает  $10^6$ .

В следующих  $n$  строках перечисляются стоимости яблок, находящиеся в клетках поля, а именно, в  $i$ -й из этих строк содержатся целые числа  $w_{i1}, w_{i2}, \dots, w_{im}$  ( $1 \leq w_{ij} \leq 10^9$ , гарантируется что для любых  $1 \leq i \leq n$  и  $1 \leq j \leq m$  выполняется неравенство  $w_{ij} < w_{a_r a_c} + w_{b_r b_c}$ ).

### Формат выходных данных

Для каждого тестового примера выведите строку, состоящую из символов «U», «D», «L», «R»: последовательность ходов змейки, выполняя которую её голова попадает из клетки  $(a_r, a_c)$  в клетку  $(b_r, b_c)$ , не посещает клетку, в которой уже находится змейка, и набирает больше, чем  $\frac{1}{2}W$  очков.

## Пример

стандартный ввод	стандартный вывод
2	RD
2 2 1 1 2 2	RRDL
1 9	
1 9	
2 4 1 2 2 3	
2 1 5 6	
3 4 8 7	

## Задача D. Ехаб и еще одна очередная задача на xor

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

### Это интерактивная задача!

Ехаб играет в игру с Лагги. Ехаб имеет два загаданных числа  $(a, b)$ . Лагги может сказать пару чисел  $(c, d)$  и Ехаб ответит:

- 1 если  $a \oplus c > b \oplus d$ .
- 0 если  $a \oplus c = b \oplus d$ .
- -1 если  $a \oplus c < b \oplus d$ .

Операция  $a \oplus b$  обозначает операцию побитовое исключающее «ИЛИ» чисел  $a$  и  $b$ .

Лагги нужно угадать  $(a, b)$  **не более, чем за 62 вопроса**. Вам предлагается сыграть в эту игру. Вы играете за Лагги, а программа жюри - за Ехаба.

**Гарантируется, что  $0 \leq a, b < 2^{30}$ .**

### Формат входных данных

См. протокол взаимодействия.

### Формат выходных данных

Чтобы вывести ответ, выведите `"! a b"` (без кавычек). **Не забудьте сбросить буфер вывода после того, как выведете ответ.**

### Протокол взаимодействия

Чтобы задать вопрос, выведите `"? c d"` (без кавычек).  $c$  и  $d$  должны быть неотрицательными целыми числами, меньшими  $2^{30}$ . **Не забудьте сбросить буфер вывода после того, как зададите вопрос.**

После каждого вопроса вы должны считать ответ. Если программа жюри отвечает числом `-2`, это значит, что ваша программа задала больше, чем 62 запроса и должна завершиться.

Чтобы сбросить буфер вывода вы можете использовать:

- `fflush(stdout)` в C++.
- `System.out.flush()` в Java.
- `stdout.flush()` в Python.
- `flush(output)` в Pascal.
- Прибегните к документации других языков.

### Пример

стандартный ввод	стандартный вывод
1	? 2 1
-1	? 1 2
0	? 2 0
	! 3 1

### Замечание

В примере из условия:

Загаданные числа:  $a = 3, b = 1$ .

В первом вопросе:  $3 \oplus 2 = 1$  и  $1 \oplus 1 = 0$ , и ответ равен 1.

Во втором вопросе:  $3 \oplus 1 = 2$  и  $1 \oplus 2 = 3$ , и ответ равен -1.

В третьем вопросе:  $3 \oplus 2 = 1$  и  $1 \oplus 0 = 1$ , и ответ равен 0.

После этого программа выводит ответ и завершается.

## Задача E. 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 скупать 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	RRDILLUR DRU
1 2 3 1 3	DRRULR



## Задача F. Bill Restoration

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

**Это задача с двойным запуском.** Ваше решение будет запущено два раза.

Члены жюри олимпиады совершили в Иннополисе крупную покупку. Для удобства весь чек оплатил один человек, а остальные должны были после вернуть свою долю. Чек, разумеется, сразу был потерян, но сохранилась сделанная для истории фотография, к сожалению, довольно плохого качества.

На ней не только не видны отдельные позиции, вошедшие в чек, но и в строке с суммой оказались смазаны  $k$  цифр подряд. Тем не менее, члены жюри, хоть и с большим трудом, справились восстановить информацию о покупке. Вам предстоит проэмулировать этот сценарий.

При **первом запуске** вам дается целое число  $A$ , состоящее ровно из  $n$  цифр — суммарная стоимость покупки, а также число  $k$  — количество подряд идущих цифр, которые необходимо будет «замазать». Гарантируется, что  $n \geq 10^k + 10$ . Вы должны выбрать ровно  $k$  подряд идущих цифр числа и заменить их на символы '#’.

При **втором запуске** ваша программа получает на вход строку, которая была получена при первом запуске заменой  $k$  подряд идущих цифр на '#’. Ваша цель — целиком восстановить исходное число  $A$ , то есть вывести, какие  $k$  цифр «спрятаны» за символами решетки.

### Формат входных данных

В этой задаче используется мультитест, каждый тест содержит несколько наборов входных данных.

При первом запуске первая строка ввода содержит через пробел число 1 и два целых числа  $t$  и  $k$  — количество наборов входных данных и количество цифр, которые необходимо закрыть в каждом из них ( $1 \leq k \leq 6$ ). В  $i$ -й из следующих  $t$  строк дана строка  $A$  для  $i$ -го набора входных данных, состоящая из цифр (символы от '0' до '9') и задающая сумму чека ( $10^k + 10 \leq |A| \leq 10^6 + 20$ ).

Гарантируется, что сумма  $|A|$  по всем наборам входных данных в одном тесте не превосходит  $2 \cdot 10^6 + 10^5$ .

При втором запуске первая строка содержит число 2 и целое число  $t$  — количество наборов входных данных. В следующих  $t$  строках даны строки  $A'$  для всех наборов входных данных. Каждая  $A'$  состоит из символов от '0' до '9' или символов '#’ и была получена при первом запуске.

**Между двумя запусками наборы входных данных могут быть перемешаны.**

### Формат выходных данных

При первом запуске для каждого набора входных данных на отдельной строке выведите целое число  $l$  от 1 до  $n - k + 1$  — позицию начала отрезка цифр длины  $k$ , который будет заменен на '#’.

При втором запуске для каждого набора входных данных выведите в отдельной строке строку длины ровно  $k$ , состоящую из цифр — последовательность, которая была спрятана за отрезком из '#’. Ответы во втором запуске следует выводить в том же порядке, в котором строки следуют во входных данных для второго запуска.

### Протокол взаимодействия

Во избежание получения некорректных вердиктов вроде `Idleness Limit Exceeded` или `Security Violation` заканчивайте вывод каждой строки символом перевода строки ('`\n`') и сбросом буфера потока вывода (`cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python).

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
0	–	примеры из условия		полная
1	10	$k = 1, t = 1,$ '1' ≤ $A_i$ ≤ '2' для всех $i$ и $A$ сгенерирована равномерно		полная
2	13	$k = 1,$ '0' ≤ $A_i$ ≤ '2' для всех $i$	1	первая ошибка
3	17	$k = 1, '0' ≤ A_i ≤ '7'$ для всех $i$	1, 2	первая ошибка
4	17	$k ≤ 2, '0' ≤ A_i ≤ '7'$ для всех $i$	1 – 3	первая ошибка
5	21	'0' ≤ $A_i$ ≤ '7' для всех $i$	1 – 4	первая ошибка
6	22	без дополнительных ограничений	0 – 5	первая ошибка

### Примеры

стандартный ввод	стандартный вывод	Пояснение
1 2 1 12345678901234567890 11111111111111111111	20 11	Первый запуск
2 2 1234567890123456789# 1111111111#111111111	0 1	Второй запуск

стандартный ввод	стандартный вывод	Пояснение
1 1 2 100000000020000000003000000000 400000000050000000005999999999 499999999939999999992999999999 19999999995555555555	51	Первый запуск
2 1 100000000020000000003000000000 40000000005000000000##99999999 499999999939999999992999999999 19999999995555555555	59	Второй запуск

## Задача G. Задача о сбалансированной раскраске

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

Даны два подвешенных дерева из  $n$  и  $m$  вершин, соответственно.

- Корень первого дерева имеет номер  $n$ , а  $i$ -я вершина ( $1 \leq i < n$ ) имеет предка  $p_i$ .
- Корень второго дерева имеет номер  $m$ , а  $i$ -я вершина ( $1 \leq i < m$ ) имеет предка  $q_i$ .

В обоих деревьях содержится ровно  $k$  листьев, они имеют номера  $1, 2, \dots, k$ . Необходимо раскрасить числа от 1 до  $k$  в красный и синий цвет таким образом, чтобы для каждого из деревьев выполнялось условие: в поддереве каждой вершины количество красных листьев отличается от количества синих не более чем на единицу.

Если подходящих раскрасок чисел несколько, то вы можете вывести любую подходящую. Гарантируется, что для всех наборов входных данных такая раскраска существует.

### Формат входных данных

В первой строке указано число  $t$  ( $1 \leq t \leq 1000$ ) – количество наборов входных данных.

В первой строке набора указаны числа  $n$  и  $m$  ( $3 \leq n, m \leq 100\,000$ ).

Во второй строке набора указаны числа  $p_1, \dots, p_{n-1}$  ( $i < p_i \leq n$ ).

В третьей строке набора указаны числа  $q_1, \dots, q_{m-1}$  ( $i < q_i \leq m$ ).

Гарантируется, что деревья удовлетворяют условию задачи, а также, что сумма  $n + m$  по всем наборам входных данных не превышает 200 000.

### Формат выходных данных

Для каждого набора входных данных выведите строку длины  $k$ , состоящую из символов R и B – искомую раскраску чисел  $1, \dots, k$ .

### Пример

стандартный ввод	стандартный вывод
2	BRRB
7 7	BRB
5 5 6 6 7 7	
5 6 5 6 7 7	
5 4	
4 4 5 5	
4 4 4	

## Задача Н. Странный прибор

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

### Это интерактивная задача.

Вася обожает решать загадки и разгадывать головоломки. На этот раз он нашел странный прибор и хочет выяснить принцип его работы.

Этот прибор зашифрован с помощью дерева (связного неориентированного графа без циклов), состоящего из  $n$  вершин, пронумерованных целыми числами от 1 до  $n$ . Чтобы решить головоломку надо угадать это дерево.

К счастью прибор умеет выполнять одну операцию, исходя из которой надо разгадать его шифр. Можно ввести в прибор последовательность  $d_1, d_2, \dots, d_n$  целых неотрицательных чисел. На приборе есть  $n$  лампочек,  $i$ -я из которых отвечает за  $i$ -ю вершину дерева-шифра. Для всех  $i$  из этих лампочек  $i$ -я загорится, если существует такая вершина дерева-шифра с номером  $j \neq i$ , что  $dist(i, j) \leq d_j$ . Здесь  $dist(i, j)$  обозначает расстояние между вершинами  $i$  и  $j$  в дереве-шифре, то есть количество ребер в простом пути между вершинами  $i$  и  $j$ .

Вася хочет за  $\leq 80$  операций с прибором решить головоломку и угадать дерево-шифр. Помогите ему!

### Протокол взаимодействия

В начале вашей программе вводится единственное целое число  $n$  — количество вершин в дереве-шифре прибора ( $2 \leq n \leq 1000$ ).

Далее вы можете выполнять операции в следующем формате. Сначала выведите символ “?” (без кавычек) и за ним  $n$  целых чисел  $d_1, d_2, \dots, d_n$ , разделенных пробелами. Заметьте, что в операциях для всех  $i$  должно быть выполнено неравенство  $0 \leq d_i < n$ . В ответ будет выведена строка  $s$  длины  $n$ , состоящая из символов “0” и “1” (без кавычек). Для всех  $i$  символ  $s_i$  будет равен “0”, если у прибора не включилась лампочка, соответствующая  $i$ -й вершине дерева-шифра и “1”, иначе.

После нескольких запросов вы должны вывести угаданное дерево. Для этого в первой строке выведите единственный символ “!” (без кавычек). В следующих  $n - 1$  строках выведите по 2 целых числа  $a_i, b_i$  — номера вершин, соединяющих  $i$ -е ребро дерева. Выведенные числа должны удовлетворять условиям  $1 \leq a_i, b_i \leq n$  и  $a_i \neq b_i$ . Эти ребра должны образовывать дерево, совпадающее с загаданным. Выводить ребра можно в любом порядке. После этого ваша программа должна завершиться.

Гарантируется, что в каждом тесте дерево-шифр будет зафиксировано заранее и не будет меняться в зависимости от выполняемых операций.

Ваша программа может выполнить от 0 до 80 операций с прибором и после этого сообщить дерево, совпадающее с загаданным.

Если ваша программа сделает больше 80 операций, то она может получить любой вердикт, потому что будет считывать данные из закрытого потока ввода. Также, если ваша программа сделает операцию или выведет ответ в неверном формате, она может получить любой вердикт. **Будьте внимательны.**

**Не забудьте сбрасывать буфер вывода после того, как выведете данные для операции или ответ.**

Чтобы сбросить буфер вывода вы можете использовать:

- `fflush(stdout)` в C++.
- `System.out.flush()` в Java.
- `stdout.flush()` в Python.
- `flush(output)` в Pascal.
- Прибегните к документации других языков.

## Пример

стандартный ввод	стандартный вывод
5	? 0 0 0 0 0
00000	? 1 1 2 0 2
11011	? 0 0 0 1 0
11100	? 0 1 0 0 1
10010	!
	4 2
	1 5
	3 4
	4 1

## Замечание

Таблица попарных расстояний между вершинами выглядит так:

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	0	2	2	1	1
<b>2</b>	2	0	2	1	3
<b>3</b>	2	2	0	1	3
<b>4</b>	1	1	1	0	2
<b>5</b>	1	3	3	2	0

- Если сделать операцию, в которой  $d = [0, 0, 0, 0, 0]$ , то ни одна лампочка не загорится, потому что  $dist(i, j) > 0$  при всех  $i \neq j$ .
- Если сделать операцию, в которой  $d = [1, 1, 2, 0, 2]$ , то загорятся все лампочки, кроме лампочки, соответствующей вершине дерева-шифра с номером 3. Например, лампочка, соответствующая вершине с номером 1 загорится, потому что  $dist(1, 5) = 1 \leq 2 = d_5$ .
- Если сделать операцию, в которой  $d = [0, 0, 0, 1, 0]$ , то загорятся все лампочки, кроме лампочек, соответствующих вершинам дерева-шифра с номерами 4 и 5.
- Если сделать операцию, в которой  $d = [0, 1, 0, 0, 1]$ , то загорятся только лампочки, соответствующие вершинам дерева-шифра с номерами 1 и 4.

## Задача I. Anime Creator

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

Это интерактивная задача.

Великий сочинитель исекаев Поликарп Монокарпович недавно посмотрел аниме Наруто, оно ему очень понравилось и он решил написать своего Наруто. В его версии аниме «Наруто 2023» изначально есть  $N$  персонажей, которых мы будем обозначать, как  $\alpha_1, \alpha_2, \dots, \alpha_n$ . Из-за сложной механики ведения боя, силы персонажей не транзитивны (то есть, если персонаж  $A$  сильнее  $B$ , а  $B$  сильнее  $C$ , то из этого не следует, что  $A$  сильнее  $C$ ). Более того, в первых сериях он продемонстрировал, что  $\alpha_1$  сильнее  $\alpha_2$ ,  $\alpha_2$  сильнее  $\alpha_3$ ,  $\dots$ ,  $\alpha_n$  сильнее  $\alpha_1$ .

Тем не менее после этого у Поликарпа не было идей для дальнейшего сюжета, в то время как аниме стало очень популярным. Поэтому он позвал вас на помощь! Каждая серия может быть, как флешбеком, так и относиться к текущему таймлайну, в каждой серии происходит бой между парой персонажей. Разница между флешбеками и основным таймлайном в том, что после боя в серии из текущего таймлайна более слабый персонаж умирает (и не может появляться в последующих сериях), а после флешбека оба остаются в живых, при этом побеждает всё равно более сильный. Таким образом, каждая серия описывается тройкой чисел  $(i, j, t)$ , где  $i$  и  $j$  – номера сражающихся персонажей, а  $t$  – индикатор, является ли серия флешбеком.

У Поликарпа есть свое мнение насчет каждой пары персонажей, которое не меняется со временем. Если в каком-то флешбеке или каких-то первых сериях  $\alpha_i$  оказался сильнее  $\alpha_j$ , то так будет, как в дальнейших флешбеках, так и в боях во время основного таймлайна.

Критикам аниме будет казаться интересным, если нельзя разбить всех живых персонажей на две группы  $A$  и  $B$  такие, что в показанных сериях ни один персонаж из группы  $B$  не победил ни одного персонажа из группы  $A$  (причём считается победа как во флешбеке, так и в серии из основного таймлайна).

Ваша цель заключается в том, чтобы показав не более, чем  $2N$  флешбеков, уменьшить количество персонажей до 3, но при этом если в какой-то момент времени аниме окажется неинтересным, то вас гарантированно уволят с комментарием «Wrong Answer».

### Протокол взаимодействия

В самом начале вашей программе будет доступно единственное число  $t$  на вход – количество наборов тестовых данных.

В начале обработки каждого набора тестовых данных на ввод будет доступно единственное число  $n$  ( $4 \leq n \leq 500$ ) – количество персонажей. Затем вы сможете предлагать серии, выводя строки вида  $i j t$  ( $1 \leq i, j \leq n$ ,  $0 \leq t \leq 1$ ,  $i \neq j$ ), где  $t = 1$ , если серия является флешбеком и  $t = 0$  в ином случае. После этого на ввод будет доступно единственное число  $x$  – номер победившего персонажа. После боя, в результате которого останутся единственные 3 выживших персонажа, обработка текущего тестового случая должна быть немедленно завершена.

Гарантируется, что сумма  $n$  по всем тестам не превышает  $10^4$ .

Если в какой-то момент времени программа жюри вывела  $-1$ , то вы должны немедленно завершить работу своей программы, иначе жюри не гарантирует, какой именно вердикт она получит.

Не забывайте после каждого запроса выводить символ перевода строки, а также сбрасывать буфер вывода командой `flush`.

## Пример

стандартный ввод	стандартный вывод
1	
4	
3	1 3 1
3	3 4 0

## Задача J. Мистер Галактика

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

Галактика ВМШ-34 известна своими интригующими конкурсами, один из них носит название «Мистер Галактика». Традиционно в последнем этапе участвуют три конкурсанта, которые до этого прошли самые сложные испытания и заслужили свое место в финале. В этот раз финалистами оказались Августор, Балдимор и Цейзос. Процесс выбора победителя конкурса достаточно прост: жители каждой планеты галактики коллективно голосуют за одного конкурсанта, при этом жители планеты могут коллективно изменить свой голос в любой момент конкурса. За организацию конкурса отвечает не менее известный Межгалактический Комитет Достоверности.

В голосовании участвует  $n$  планет и исходно  $i$ -я планета голосует за кандидата  $s_i$ .

Назовем последовательность подряд идущих голосов планет *достоверной*, если в ней **не найдется** трех планет с такими номерами  $i, j, k$  ( $1 \leq i < j < k \leq n$ ), которые удовлетворяют следующим условиям:  $i + k = 2 \cdot j$  и  $s_i = s_j = s_k$ .

*Число достоверности* будем называть количество *достоверных* последовательностей голосов с длиной не менее 3. Две последовательности считаются различными, если хотя бы одна планета входит в одну последовательность, но не входит в другую.

Голосование длится  $q$  часов и при этом известно, что под конец  $i$ -го часа  $x_i$ -я планета коллективным решением изменяет свой голос в пользу кандидата  $d_i$ .

Межгалактический Комитет Достоверности очень интересуется динамикой изменения *числа достоверности* на протяжении всего конкурса, поскольку по их мнению по этим данным можно выявить аномалии и в случае необходимости дисквалифицировать нечестных конкурсантов. Поэтому они попросили Вас, независимого наблюдателя из другой галактики, помочь им в вычислении *числа достоверности* после каждого изменения голосов.

### Формат входных данных

В первой входной строке содержится два натуральных числа  $n, q$  — количество планет в галактике ВМШ-34 и длительность голосования конкурса.

Следующая входная строка содержит строку  $s$ , состоящую из  $n$  символов  $s_i$  — исходные голоса планет за конкурсантов. Символ на позиции  $i$  обозначает голос планеты под номером  $i$  ('a' — голос планеты был отдан за Августора, 'b' — голос планеты был отдан за Балдимора, 'c' — голос планеты был отдан за Цейзоса).

Последующие  $q$  строк содержат натуральное число  $x_i$  и символ  $d_i$  — кандидат за которого голосует планета под номером  $x_i$  после  $i$ -го часа.

$$1 \leq n \leq 10^5$$

$$1 \leq q \leq 2 \cdot 10^5$$

$$d_i, s_i \in \{a, b, c\}$$

$$1 \leq x_i \leq n$$

### Формат выходных данных

На каждой из  $q$  строк выведите целое число — число достоверности после  $i$ -го часа.



## Примеры

стандартный ввод	стандартный вывод
6 3	8
abacab	4
2 c	10
3 c	
4 a	
10 7	16
aaabbbabaa	20
1 b	19
4 a	20
6 a	16
9 b	20
10 b	19
9 a	
10 a	

## Замечание

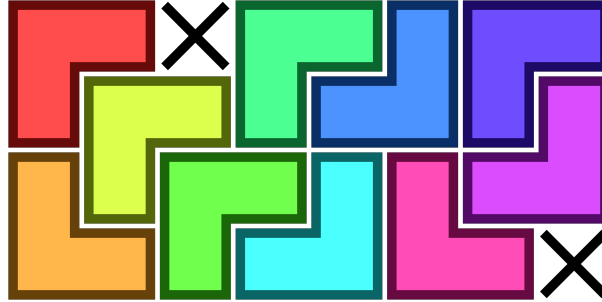
Рассмотрим первый запрос из первого примера. Галактика ВМШ-34 состоит из 6 планет. После первого изменения голоса планеты под номером 2 последовательность голосов выглядит следующим образом — 'асасаб'. Достоверными последовательностями после этого изменения являются:

- асасаб;
- асасаб;
- асасаб;
- асасаб;
- асасаб;
- асасаб;
- асасаб;
- асасаб;

## Задача К. Таблица и уголки

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

Саше подарили на день рождения таблицу  $n \times m$  и бесконечное количество уголков из трех клеток. Задача для Саши — уложить как можно больше уголков в заданную таблицу.



Пример укладки уголков в таблицу  $4 \times 8$ .

### Формат входных данных

В единственной строке указана пара чисел  $n$  и  $m$  — размеры таблицы.

$$1 \leq n, m \leq 10\,000$$

$$1 \leq n \cdot m \leq 100\,000$$

### Формат выходных данных

Выведите  $n$  строк длины  $m$ , описывающих требуемую укладку. При этом:

- Символ «.» обозначает клетку, которая не покрыта ни одним уголком.
- Уголки следует обозначать заглавными буквами латинского алфавита. Все три клетки одного уголка должны быть обозначены одной и той же буквой.
- Два уголка, которые имеют общую сторону, должны быть обозначены различными буквами.

Если подходящих укладок несколько — выведите любую из них.

### Примеры

стандартный ввод	стандартный вывод
4 4	AABB ACCB DC.E DDEE
4 8	AA.EEGHN ACCEGGHI BCDDFJII BBDFFJJ.
1 4	....
3 4	AABB ABAB BBA