

## Задача А. Проективное расстояние

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

Рассмотрим прямоугольное поле, состоящее из  $w \times h$  квадратных клеток. В клетке  $A$  стоит фишка. За один ход можно подвинуть фишку в любую соседнюю по стороне клетку. Сколько ходов потребуется, чтобы добраться до клетки  $B$ ?

Важное дополнение: края поля склеены по принципу проективной плоскости. А именно,  $i$ -я сверху клетка левого края — соседняя с  $i$ -й снизу клеткой правого края, а  $j$ -я слева клетка верхнего края — соседняя с  $j$ -й справа клеткой нижнего края (для всех  $i$  и  $j$ , для которых существуют эти клетки).

### Формат входных данных

В первой строке заданы два целых числа  $w$  и  $h$  — ширина и высота поля ( $1 \leq w, h \leq 10^8$ ).

Во второй строке заданы два целых числа  $x_A$  и  $y_A$  — столбец и строка клетки  $A$ .

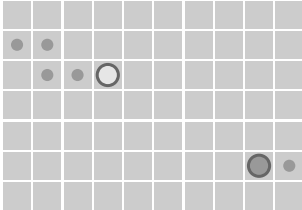
В третьей строке заданы два целых числа  $x_B$  и  $y_B$  — столбец и строка клетки  $B$ .

Столбцы пронумерованы от 1 до  $w$  слева направо, а строки — от 1 до  $h$  сверху вниз.

### Формат выходных данных

Выведите одно целое число: минимальное количество ходов, за которое можно добраться из клетки  $A$  в клетку  $B$ .

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>иллюстрация</i>
10 7 9 6 4 3	6	

### Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (40 баллов)  $1 \leq w, h \leq 100$ .

Во второй подзадаче (60 баллов) дополнительных ограничений нет.

## Задача В. Максимальный отрезок на кольце

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

Задан массив, состоящий из целых чисел. Как выбрать отрезок массива, на котором сумма чисел максимальна?

Важное дополнение: массив записан на окружности так, что после последнего элемента следует первый. Отрезок на таком массиве — дуга этой окружности. В частности, выбранный отрезок может быть пустым или покрывать весь массив.

### Формат входных данных

В первой строке задано целое число  $n$  — размер массива ( $1 \leq n \leq 300\,000$ ).

Во второй строке заданы  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — элементы массива ( $|a_i| \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число: максимальную сумму на отрезке закольцованного массива.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
6 -2 4 -10 8 -3 5	12	<u>...</u> -2 4 -10 8 -3 5 <u>...</u>

### Система оценки

В этой задаче три подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (20 баллов)  $1 \leq n \leq 100$ .

Во второй подзадаче (30 баллов)  $1 \leq n \leq 5000$ .

В третьей подзадаче (50 баллов) дополнительных ограничений нет.

## Задача С. Поиск подстроки в циклической строке

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

Даны две строки,  $s$  и  $p$ . Сколько раз  $p$  встречается в  $s$  как подстрока?

Важное дополнение: строка  $s$  — не простая, а циклическая. Циклическую строку можно представить себе так: запишем её на бумажной ленте, после чего склеим концы ленты так, чтобы после последней буквы  $s$  следовала первая её буква.

Пусть  $s$  — циклическая строка из  $n$  букв, а  $p$  — обычная строка из  $k$  букв. Определим формально, сколько раз  $p$  встречается в  $s$ . Для каждого  $i = 1, 2, \dots, n$  начнём с  $i$ -й буквы строки  $s$  и прочитаем ровно  $k$  букв циклической строки. Если получилась в точности строка  $p$  — будем говорить, что она встречается в  $s$  как подстрока, начиная с  $i$ -й буквы. Количество таких  $i$  — это и есть количество вхождений  $p$  в  $s$ .

### Формат входных данных

В первой строке задана циклическая строка  $s$ , а во второй — обычная строка  $p$ . Обе строки состоят из маленьких английских букв и имеют длину от 1 до  $10^6$  символов.

### Формат выходных данных

Выведите одно целое число: сколько раз  $p$  встречается в циклической строке  $s$  как подстрока.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
mamba am	2	...mambamamba... 1: am no 2: am yes 3: am no 4: am no 5: am yes
aba abaab	1	...abaabaaba... 1: abaab yes 2: abaab no 3: abaab no
ababab baba	3	...abababababab... 1: baba no 2: baba yes 3: baba no 4: baba yes 5: baba no 6: baba yes

### Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (35 баллов) обе заданные строки состоят не более чем из 1000 букв.

Во второй подзадаче (65 баллов) дополнительных ограничений нет.

## Задача D. Бесконечные периодические строки

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

Даны три целых числа:  $a$ ,  $b$  и  $c$ . Сколько существует строк, которые имеют период  $a$ , имеют период  $b$ , но не имеют периода  $c$ ?

Важное дополнение: все строки в этой задаче — двоичные и бесконечные **в обе стороны**.

Можно зафиксировать какую-то позицию в строке, объявить её нулевой и пронумеровать все позиции, начиная с неё: справа положительными числами, а слева отрицательными. Например, рассмотрим строку  $s = \dots 0000011111\dots$ , в которой слева — бесконечное количество нулей, а справа — бесконечное количество единиц. Если мы начнём нумерацию позиций с позиции самой левой единицы, то  $s_0 = s_1 = s_2 = s_3 = \dots = 1$ , а  $s_{-1} = s_{-2} = s_{-3} = \dots = 0$ .

Две строки, отличающиеся только нумерацией позиций, будем считать одинаковыми. Например, если строка  $s$  задаётся как  $s_i = 1$  при  $i \geq 0$  и  $s_i = 0$  при  $i < 0$ , а строка  $t$  — как  $t_i = 1$  при  $i \geq -5$  и  $t_i = 0$  при  $i < -5$ , то эти строки одинаковые: сдвинув нумерацию на 5 позиций, мы из одной строки можем получить другую. А вот строка  $r$ , в которой  $r_i = 1$  при  $i \leq 0$  и  $r_i = 0$  при  $i > 0$ , с ними не совпадает.

Будем говорить, что строка имеет период  $p$ , если на любых двух позициях, номера которых отличаются на  $p$ , стоят одинаковые символы. Например, посмотрим на строку  $\dots 01010101010101\dots$ , в которой соседи любой единицы — нули, а соседи любого нуля — единицы. Эта строка имеет период 2, имеет также период 4, но не имеет периода 3.

### Формат входных данных

В первой строке заданы три целых числа  $a$ ,  $b$  и  $c$  ( $1 \leq a, b, c \leq 60$ ).

### Формат выходных данных

Выведите одно целое число: сколько существует различных бесконечных в обе стороны двоичных строк, которые имеют период  $a$ , имеют период  $b$ , но не имеют периода  $c$ .

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
6 3 2	2	$\dots 011011011011\dots$ $\dots 100100100100\dots$
4 4 4	0	
6 12 4	11	$\dots 000001000001\dots$ $\dots 000011000011\dots$ $\dots 000101000101\dots$ $\dots 000111000111\dots$ $\dots 001001001001\dots$ $\dots 001011001011\dots$ $\dots 001101001101\dots$ $\dots 001111001111\dots$ $\dots 010111010111\dots$ $\dots 011011011011\dots$ $\dots 011111011111\dots$

### Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (30 баллов) числа  $a$ ,  $b$  и  $c$  не превосходят 20.

Во второй подзадаче (70 баллов) дополнительных ограничений нет.

## Задача Е. Кольцевой калькулятор

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

Дисплей калькулятора вмещает  $n$  десятичных цифр. Что получится, если на этом калькуляторе посчитать  $x^y$  ( $x$  в степени  $y$ )?

Важное дополнение: это кольцевой калькулятор. Дисплей калькулятора зациклен так, что слева от самого старшего ( $n$ -го справа) разряда следует младший. Действия, которые при обычных вычислениях потребовали бы больше  $n$  цифр, вместо этого используют  $n$  разрядов по кругу.

Определим формально действия на этом калькуляторе. Пусть  $a$  — число на дисплее.

- Прибавление к  $a$  единицы: если  $a$  состоит из всех девяток, то оно превращается в 1, во всех остальных случаях  $a$  превращается в  $a + 1$ .
- Прибавление к  $a$  числа  $b$ : начнём с  $a$  и  $b$  раз прибавим единицу.
- Умножение  $a$  на число  $c$ : начнём с нуля и  $c$  раз прибавим число  $a$ .
- Возведение  $a$  в степень  $d$ : начнём с единицы и  $d$  раз умножим на число  $a$ .

Можно показать, что верно также более наглядное пояснение. Чтобы получить результат арифметических действий — например, сложения или умножения — можно выполнять их как обычно. Но после этого, если в ответе больше  $n$  цифр, следует прибавить лишние цифры в соответствующие разряды.

Например, пусть  $n = 4$ . При обычном сложении  $9999 + 1$  получается число 10 000. Но на кольцевом калькуляторе единица, которая должна была попасть в пятый справа разряд, попадает вместо этого в младший разряд. Итоговый ответ равен 1.

Другой пример: при обычном умножении  $321 \cdot 321$  получается число 103 041. Но теперь, если на дисплее всего  $n = 4$  цифры — чтобы получить ответ, нужно прибавить к числу из  $n$  младших разрядов (3041) число из старших разрядов (10). Итоговый ответ равен 3051.

### Формат входных данных

В первой строке заданы три целых числа:  $n$ ,  $x$  и  $y$  ( $1 \leq n \leq 9$ ;  $1 \leq x, y < 10^n$ ).

### Формат выходных данных

Выведите одно целое число: результат вычисления  $x^y$  на кольцевом калькуляторе с  $n$  десятичными цифрами на дисплее. Допускается вывод лишних нулей в начале числа, если общее количество цифр в выведенном ответе не превосходит  $n$ .

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
4 321 2	3051	$\begin{array}{r l}  0321 &  3041 \\ * &  0321 & + &  0010 \\ --- ---- & -- ---- \\ &  0321 & &  3051 \\ + &  6420 \\ + & 9 6300 \\ --- ---- \\ & 10 3041 \end{array}$

### Система оценки

В этой задаче один пример и 50 основных тестов. Каждый основной тест оценивается отдельно и даёт 2 балла.

## Задача F. Переворачивание дуг

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

Задана изначальная строка, а также последовательность операций: переворачивание частей строки. Какая строка получится после всех операций?

Важное дополнение: строка записана на ленте, склеенной в виде кольца — так, что после последней буквы строки следует первая. Позиции на ленте пронумерованы подряд целыми числами от 1 до  $n$ , где  $n$  — длина строки. Изначальная строка записана от позиции 1 до позиции  $n$ .

Каждая операция — переворачивание дуги. Операция задаётся двумя позициями:  $l$  и  $r$ . На кольце выделяется дуга, состоящая из идущих подряд позиций от  $l$  до  $r$ :  $l, l + 1, \dots, r - 1, r$  (если  $l > r$ , дуга содержит последнюю и первую позиции). Все буквы на этой дуге переставляются в обратном порядке: первая буква на дуге меняется местами с последней, вторая — с предпоследней, и так далее.

Например, пусть  $n = 5$ , и изначальная строка равна «acros». После операции с параметрами  $l = 1$  и  $r = 4$  перевернётся дуга, состоящая из позиций 1, 2, 3, 4, и строка «acros» превратится в «orcas». После следующей операции с параметрами  $l = 5$  и  $r = 2$  перевернётся дуга, состоящая из позиций 5, 1, 2, и строка «orcas» превратится в «oscar».

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $q$ : длина строки и количество операций ( $1 \leq n, q \leq 200\,000$ ). Во второй строке задана сама строка, состоящая из  $n$  маленьких английских букв. Каждая из следующих  $q$  строк содержит параметры очередной операции: два целых числа  $l$  и  $r$  ( $1 \leq l, r \leq n$ ).

### Формат выходных данных

Выведите строку, которая получится после всех операций.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
5 2 acros 1 4 5 2	oscar	1, 2, 3, 4: 1234. -> 4321. acros -> orc <u>as</u>  5, 1, 2: 12..5 -> 15..2 orc <u>as</u> -> oscar
6 2 purest 3 2 1 1	erupts	3, 4, 5, 6, 1, 2: 123456 -> 432165 purest -> erupts  1: 1..... -> 1..... erupts -> erupts

### Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (40 баллов) числа  $n$  и  $q$  не превосходят 1000.

Во второй подзадаче (60 баллов) дополнительных ограничений нет.

## Задача Г. Игра с дугами

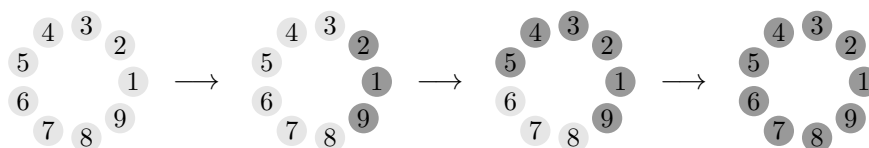
Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

На ленте отмечены в ряд  $n$  белых точек. Марша и Билл играют в игру, делая ходы по очереди. Первой ходит Марша. Ход — выбрать  $k$  точек подряд так, чтобы они все были белыми, и перекрасить их в чёрный. Проигрывает тот, кто не может ходить. Кто выигрывает при правильной игре?

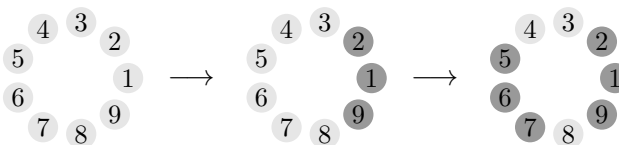
Важное дополнение: концы ленты склеены так, что после последней отмеченной точки следует первая. Последняя и первая точки считаются стоящими подряд.

Например, пусть  $n = 9$  и  $k = 3$ . Пронумеруем точки по порядку числами от 1 до 9. Тогда первым ходом Марша может, например, выбрать точки 9, 1, 2 и перекрасить их в чёрный.

Теперь, если Билл выберет точки 3, 4, 5 и перекрасит их в чёрный, то Марша выберет 6, 7, 8, перекрасит их в чёрный и выиграет:



Если же Билл выберет точки 5, 6, 7 и перекрасит их в чёрный, то Марша не сможет сделать ход и проиграет:



### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $k$ : количество точек и число для хода ( $1 \leq n \leq 50\,000$ ;  $1 \leq k \leq 10$ ).

### Формат выходных данных

Если при правильной игре выигрывает Марша, выведите «Marsha».

Если при правильной игре выигрывает Билл, выведите «Bill».

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
9 3	Bill
7 2	Marsha

### Система оценки

В этой задаче три подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (30 баллов) число  $n$  не превосходит 20.

Во второй подзадаче (60 баллов) число  $n$  не превосходит 500.

В третьей подзадаче (10 баллов) дополнительных ограничений нет.

## Задача Н. Пересылка через репитер

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

*Это задача с двойным запуском.*

Вика хочет передать Серёже строку из  $n$  двоичных цифр. Для передачи используется *репитер*. Вика может послать не более  $n + 100$  двоичных цифр на передачу. Серёжа должен, получив эти цифры, восстановить исходную строку.

Важное дополнение: репитер выдаёт не передаваемую строку, а какой-то её циклический сдвиг. Формально, если передаваемая Викой строка равна  $t_1t_2\dots t_m$ , то репитер выбирает произвольную позицию  $p$  от 1 до  $m$ , после чего выдаёт Серёже строку  $t_pt_{p+1}\dots t_mt_1t_2\dots t_{p-1}$ .

Помогите Вике и Серёже договориться о передаче так, чтобы Серёжа, получив строку от репитера, мог восстановить исходную строку.

### Первый запуск

В этой задаче ваше решение будет запущено на каждом тесте два раза.

При первом запуске решение действует за Вика. В первой строке записано слово «send». Во второй строке задано целое число  $n$  — длина исходной строки ( $1 \leq n \leq 450\,000$ ). В третьей строке задана исходная строка  $s$ , состоящая из  $n$  двоичных цифр.

Выведите строку из двоичных цифр, которую следует передать. Количество цифр должно быть от 1 до  $n + 100$  включительно.

### Второй запуск

При втором запуске решение действует за Серёжу. В первой строке записано слово «receive». Во второй строке задано целое число  $m$  — длина строки, переданной в первом запуске. В третьей строке задан циклический сдвиг переданной строки.

Выведите исходную строку  $s$ .

Позиция  $p$  для циклического сдвига ведёт себя как случайное число, выбранное равномерно от 1 до  $m$ . При этом в каждом тесте, если два решения вывели одну и ту же строку при первом запуске, они получают одну и ту же строку при втором запуске.

### Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
send 5 01001	000000100111111
receive 15 011111100000010	01001

### Система оценки

В этой задаче две подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (50 баллов) число  $n$  не превосходит 45.

Во второй подзадаче (50 баллов) дополнительных ограничений нет.



## Задача I. Поиск на зигзаге

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

*Это интерактивная задача.*

Есть секретный массив, состоящий из  $n$  различных чисел. Позиции в массиве пронумерованы подряд целыми числами от 1 до  $n$ . Можно задавать вопрос: чему равно число на позиции  $i$ ?

Даны  $k$  различных чисел из массива. Определите позиции этих  $k$  чисел в массиве, задав не более 500 вопросов.

Важное дополнение: массив записан на ленте, концы которой склеены так, что после последнего элемента следует первый. Зацикленный массив устроен как *зигзаг*. А именно, в нём есть две различные позиции,  $p$  и  $q$ . На части ленты от позиции  $p$  до позиции  $q$  следующий элемент больше предыдущего. На части ленты от позиции  $q$  до позиции  $p$  следующий элемент меньше предыдущего. Напомним, что все числа в массиве различны, поэтому все неравенства строгие.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $k$ : размер массива и количество чисел для поиска ( $2 \leq n \leq 200\,000$ ;  $1 \leq k \leq 10$ ). Во второй строке заданы  $k$  различных чисел из массива, позиции которых нужно определить.

Массив в каждом тесте зафиксирован заранее и не меняется в процессе работы.

### Протокол взаимодействия

Чтобы узнать число на позиции  $i$  ( $1 \leq i \leq n$ ), выведите номер позиции в отдельной строке — и очистите буфер вывода, чтобы не получить вердикт IL (Idleness Limit Exceeded). После этого прочитайте в отдельной строке число  $a_i$  —  $i$ -й элемент массива ( $1 \leq a_i \leq 10^9$ , все  $a_i$  различны).

Чтобы закончить взаимодействие, выведите в отдельной строке число 0 и завершите работу программы. Решение считается верным, если вопросов (не считая последний 0) было не более 500 и каждое из  $k$  заданных во вводе чисел встретилось **хотя бы раз** среди ответов на вопросы.

Заметим, что ограничение на количество вопросов проверяется только после окончания взаимодействия.

### Пример

В примере секретный массив таков:  $a = (6, 9, 8, 5, 3, 1, 2)$ . От позиции  $p = 6$  до позиции  $q = 2$  следующий элемент больше предыдущего:  $1 < 2 < 6 < 9$ . От позиции  $q = 2$  до позиции  $p = 6$  следующий элемент меньше предыдущего:  $9 > 8 > 5 > 3 > 1$ .

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
7 3	4	$a[4] = 5$ (found 5)
1 6 5	6	$a[6] = 1$ (found 1)
5	2	$a[2] = 9$ (miss)
1	1	$a[1] = 6$ (found 6)
9	6	$a[6] = 1$ (repeat)
6	0	
1		

### Система оценки

В этой задаче три подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (10 баллов) число  $n$  не превосходит 200.

Во второй подзадаче (70 баллов) число  $n$  не превосходит 2000.

В третьей подзадаче (20 баллов) дополнительных ограничений нет.

## Задача J. Наименьшее количество инверсий

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

*Это задача с двойным запуском.*

У Вани была перестановка  $p$  размера  $n$ : последовательность из чисел  $1, 2, \dots, n$ , в которой каждое число встречается один раз. Ваня хотел бы узнать число  $x$  — количество инверсий в ней, то есть количество пар позиций  $(i, j)$ , для которых  $1 \leq i < j \leq n$  и при этом  $p_i > p_j$ .

Важное дополнение: Ваня не успел сам посчитать  $x$ , ибо судьба перестановки непроста.

Сначала в гости к Ване зашёл Влад и перенёс эту перестановку на ленту, концы которой склеены так, что после последнего элемента следует первый. Теперь неизвестно, с какого из  $n$  элементов начиналась перестановка. Тем не менее, Ваня уверен, что число  $x$  — **минимальное** из тех, что могут получиться, если разрезать ленту по какой-то из  $n$  линий между соседними числами.

Затем в гости к Ване зашёл Никита и разрезал эту ленту на два непустых куска: в каждом куске получилось от 1 до  $n - 1$  чисел. Теперь Ваня готов сначала показать вам первый кусок, а потом спрятать его, пока с ним ещё что-нибудь не случилось, и показать второй кусок.

Несмотря на все трудности, помогите Ване узнать число  $x$ .

### Первый запуск

В этой задаче ваше решение будет запущено на каждом тесте два раза.

При первом запуске решение получает первый кусок ленты. В первой строке записано слово «first». Во второй строке заданы два целых числа  $n$  и  $s$  — длина перестановки и размер первого куска ( $1 \leq s < n \leq 300\,000$ ). В третьей строке заданы  $s$  чисел  $p_1, p_2, \dots, p_s$  — первый кусок перестановки.

Выведите строку  $h$ , содержащую от 1 до 100 произвольных символов с ASCII-кодами от 32 до 126 включительно.

### Второй запуск

При втором запуске решение получает второй кусок ленты, а также строку, выведенную при первом запуске. В первой строке записано слово «second». Во второй строке заданы два целых числа  $n$  и  $t$  — длина перестановки, та же, что при первом запуске, и размер второго куска. В третьей строке заданы  $t$  чисел  $q_1, q_2, \dots, q_t$  — второй кусок перестановки. Гарантируется, что первый и второй кусок вместе образуют перестановку длины  $n$ . В четвёртой строке приведена строка  $h$ , выведенная при первом запуске.

Выведите одно целое число  $x$ : минимально возможное количество инверсий в перестановке, если она могла начинаться с любой позиции на ленте.

В каждом тесте вся перестановка зафиксирована заранее и не меняется между запусками.

### Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
first 5 3 1 5 4	!@# 1 5 4 #@!	Print a line in any format, as long as it's short: at most 100 characters.
second 5 2 3 2 !@# 1 5 4 #@!	4	1 5 4 3 2: 6 inversions 5 4 3 2 1: 10 inversions 4 3 2 1 5: 6 inversions 3 2 1 5 4: 4 inversions 2 1 5 4 3: 4 inversions

## Система оценки

В этой задаче четыре подзадачи. Чтобы получить баллы за подзадачу, нужно пройти все тесты этой подзадачи и всех предыдущих подзадач.

В первой подзадаче (25 баллов) число  $n$  не превосходит 30.

Во второй подзадаче (25 баллов) число  $n$  не превосходит 300.

В третьей подзадаче (25 баллов) число  $n$  не превосходит 3000.

В четвёртой подзадаче (25 баллов) дополнительных ограничений нет.

---