

## Задача А. Мясо и капуста

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

В комнате стоит очень длинный узкий стол, на котором слева направо расположено  $n$  продуктов животного и растительного происхождения. Чтобы не углубляться в тонкости пищевой индустрии, будем их называть кусками мяса (meat) и кочанами капусты (cabbage). Если согласно этому упрощению каждому из продуктов сопоставить букву «m» в случае мяса и «c» в случае капусты, получится строка из  $n$  символов.

Возле левого края стола встала гимнастка Алина, а возле правого — веган Ваня. Они желают съесть всю еду со стола, при этом Алина может есть только мясо, а Ваня — только капусту. При этом по правилам, принятым в этой комнате, человек может брать только ближайший продукт к его краю стола и, если он взял что-то, то он должен это сразу съесть. Эту процедуру можно повторять несколько раз: так, если три ближайших к Ваниному краю стола продукта — это три кочана капусты, то он может съесть от нуля до трёх из этих кочанов; если же ближайший к Ване продукт — мясо, то Ваня вообще не может ничего есть. Кроме того, двум людям нельзя стоять с одного края стола, но при этом два человека с разных краёв стола могут поменяться местами, если захотят. Подумав, Алина и Ваня поняли, что правила действительно позволяют им съесть всё, что лежит на столе, за какое-то число обменов, но, поскольку стол длинный, меняться местами довольно долго. Поэтому Алина и Ваня попросили найти вас наименьшее достаточное число обменов.

### Формат входных данных

В первой строке находится целое число  $n$  — количество продуктов питания на столе ( $1 \leq n \leq 200\,000$ ). Во второй строке записано  $n$  строчных латинских букв, каждая из которых — либо «m» для мяса, либо «c» для капусты. Символы указаны в том же порядке, в каком соответствующие им продукты расположены на столе.

### Формат выходных данных

Выведите целое число — наименьшее количество раз, которое могут поменяться местами Алина и Ваня, чтобы съесть со стола всё мясо и всю капусту.

### Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех групп с более строгими ограничениями, а также всех примеров.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	тесты из условия
1	22	$1 \leq n \leq 20$	0	—
2	41	$1 \leq n \leq 1000$	0, 1	—
3	37	$1 \leq n \leq 200\,000$	0, 1, 2	—

## Примеры

стандартный ввод	стандартный вывод
14 mmmmmmmmcccccc	0
10 ccmmmmmmmm	1
7 mcmcmcm	3

## Пояснения к примерам

В первом примере Алине и Ване незачем меняться местами: изначально Алине уже доступно всё мясо, а Ване — вся капуста (так как Алина стоит с левого края, а Ваня — с правого).

Во втором примере с Алининого края лежит капуста, а с Ваниного — мясо, поэтому они не могут ничего съесть. Но как только они поменяются — и Ваня встанет слева, а Алина встанет справа — они сразу смогут съесть всю еду.

В третьем примере первым делом Алина съест ближний кусок мяса. После этого Алине и Ване надо трижды повторить следующее: поменяться местами, а затем каждому из них съесть ближайший к краю продукт.

## Задача В. Глеб и задачи

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

Сегодня у Глеба важный день, так как он снова ведёт кружок по программированию. Но вот проблема: электричка задержалась, и теперь он опаздывает. Глеб очень ответственный, поэтому материалы для занятия у него были готовы заранее. Так что он просто выслал каждому ученику одну из подготовленных задач. Задачи у всех учеников различны.

Глеб отличается пунктуальностью, поэтому он записал в строку последовательность символов: какую задачу выдал первому ученику, какую второму, и так далее до последнего ученика. Задачи у Глеба пронумерованы заглавными и строчными буквами английского алфавита. Одна и та же буква в разных регистрах обозначает различные задачи.

Ученики Глеба очень умные, поэтому все справились со своими задачами до того, как преподаватель успел приехать. Так что теперь нашему герою нужно вновь выдать те же самые задачи, по одной каждому ученику. Но при этом ни один ученик не должен получить ту же задачу, которую решил до этого.

Помогите Глебу выдать ученикам задачи во второй раз, или выясните, что это невозможно.

### Формат входных данных

В единственной строке задана непустая последовательность символов без пробелов: какая задача выдана первому ученику, какая второму, и так далее до последнего ученика. Каждый из этих символов — строчная или заглавная буква английского алфавита. Гарантируется, что все символы в последовательности различны. Входные данные завершаются переводом строки.

### Формат выходных данных

В единственной строке выведите непустую последовательность символов без пробелов: какую задачу выдать первому ученику, какую второму, и так далее до последнего ученика. Это должны быть те же символы, что и во входных данных, каждый ровно по одному разу. Но они должны быть расположены в таком порядке, чтобы ни один ученик не получил ту же задачу, которую решил до этого.

Если возможных ответов несколько, можно вывести любой из них. Если же выдать задачи во второй раз, соблюдая все условия, невозможно, выведите «-1» (без кавычек).

### Система оценки

Каждый тест в этой задаче оценивается отдельно.

### Пример

стандартный ввод	стандартный вывод
abDd	badD

## Задача С. Глеб и электрокар

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Недавно Глеб всерьёз озаботился проблемами экологии: кругом говорят о глобальном потеплении, тоннах пластика в океанах, нехватке ресурсов. «Современные проблемы требуют современных решений», — подумал он. Поэтому, не желая дальше загрязнять воздух выхлопами своего автомобиля «Волга», Глеб прикупил себе новенький электрокар от компании Tesla. При этом он узнал, что без подзарядки такое средство передвижения может проехать лишь  $k$  километров.

Глеб живёт в стране, где  $n$  городов и  $m$  двухсторонних дорог, их соединяющих. Дорога с номером  $i$  связывает города  $p_i$  и  $q_i$  и имеет длину  $r_i$  километров. В дорожной сети нет дорог, соединяющих город с самим собой, а между двумя разными городами может быть не более одной дороги. Глеб живёт в городе под номером  $u$ , где воздух кажется ему слишком загрязнённым. Так что с детства его главная мечта — переехать в город  $v$ . И если на своей старенькой «Волге» наш герой никак не мог добраться до конечного пункта, то теперь, когда у него есть новенький электрокар, он всерьёз намерен осуществить свою мечту.

Так как электрокары появились относительно недавно, заправки для них есть пока только в  $l$  городах:  $a_1, a_2, \dots, a_l$ . Глеб просит помощи в осуществлении своей мечты, так что вам предстоит узнать, сможет ли он добраться до города  $v$ , и если да, то какое наименьшее количество раз придётся заправиться. Когда Глеб садится в машину в городе  $u$ , она уже полностью заряжена. Оказавшись в городе с заправкой, Глеб может как использовать её, так и проигнорировать.

### Формат входных данных

Первая строка содержит пять целых чисел:  $k$  — сколько километров может проехать электрокар без подзарядки,  $n$  — количество городов,  $m$  — количество дорог,  $u$  — номер города Глеба, и  $v$  — номер города, куда хочет добраться Глеб ( $1 \leq k \leq 500$ ,  $2 \leq n \leq 10\,000$ ,  $0 \leq m \leq 10\,000$ ,  $1 \leq u, v \leq n$ ,  $u \neq v$ ).

В следующих  $m$  строках задаются дороги. В  $i$ -й из этих строк записаны три целых числа  $p_i$ ,  $q_i$  и  $r_i$  — номера двух городов, которые соединяет очередная двухсторонняя дорога, и её длина ( $1 \leq p_i, q_i \leq n$ ,  $1 \leq r_i \leq 10^9$ ).

Следующая строка содержит целое число  $l$  — количество заправок ( $0 \leq l \leq n$ ). Наконец, последняя строка содержит  $l$  целых чисел  $a_1, a_2, \dots, a_l$  — номера городов с заправками в порядке возрастания ( $1 \leq a_1 < a_2 < \dots < a_l \leq n$ ).

### Формат выходных данных

Выведите «-1» (без кавычек), если невозможно доехать от города с номером  $u$  до города с номером  $v$ , или минимальное количество заправок, если это возможно.

### Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех групп с более строгими ограничениями, а также всех примеров.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	тесты из условия
1	20	$n \leq 7$	0	—
2	20	$n \leq 1000$	0, 1	—
3	30	заправки есть во всех городах	0	—
4	30	нет дополнительных ограничений	0, 1, 2, 3	—

## Примеры

стандартный ввод	стандартный вывод
3 3 3 1 3 1 2 3 1 3 4 2 3 3 2 2 3	1
3 3 3 1 3 1 2 2 1 3 4 2 3 2 0	-1
3 3 3 1 3 1 2 2 1 3 4 2 3 1 0	0

## Пояснения к примерам

В первом примере Глеб должен проехать из первого города в третий. При этом он не может поехать по прямой дороге, потому что заряда электрокара хватит только на три километра, а длина дороги — четыре километра. Однако Глеб может проехать через второй город и заправиться там.

Во втором примере Глеб не может доехать до третьего города, потому что ему нужно проехать четыре километра без дозаправки.

В третьем примере Глеб может проехать из первого города во второй, а затем в третий. Тогда он проедет всего три километра. Поскольку  $k = 3$ , заправляться не нужно.

## Задача D. Глеб и MTG

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

Недавно Глеб узнал про новую для себя игру — Magic: The Gathering. Раньше из карточных игр он играл только в Hearthstone, так что разнообразие механик в MTG удивило Глеба. А такие механики, как «вы не можете проиграть» и «если у вас умерло существо, отправьте карты противника на кладбище», просто шокировали его.

Глеб решил придумать свою механику: «прогрессия». А именно: для каждой карты волшебства (sorcery) в колоде посмотрим на прямой ущерб, наносимый этим волшебством противнику (если волшебство лечит игрока, это число отрицательно; если волшебство не лечит и не наносит прямого ущерба, это число равно нулю). Если все эти числа содержатся в какой-то арифметической прогрессии, состоящей из целых чисел, то здоровье игрока удваивается.

Теперь Глебу интересно, подходит ли его колода под новую механику. Помогите ему это выяснить.

### Формат входных данных

В первой строке задано одно целое число  $n$  — количество карт волшебства у Глеба в колоде ( $1 \leq n \leq 100\,000$ ). В следующей строке заданы  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-10^9 \leq a_i \leq 10^9$ ), где  $a_i$  — число, соответствующее  $i$ -му волшебству в колоде.

### Формат выходных данных

Если колода Глеба не подходит под новую механику, в первой строке выведите «NO». Иначе в первой строке выведите «YES», а во второй — такие целые числа  $s$  и  $d$ , что каждое из чисел  $a_i$  встречается в арифметической прогрессии  $s, s + d, s + 2d, s + 3d, \dots$

Выведенные числа должны быть не очень велики: они должны удовлетворять ограничениям  $-10^9 \leq s \leq 10^9$  и  $-10^9 \leq d \leq 10^9$ . Гарантируется, что если ответ существует, то существует и ответ, подходящий под эти ограничения. Из правильных ответов, подходящих под эти ограничения, разрешается вывести любой.

### Система оценки

Каждый тест в этой задаче оценивается отдельно.

### Пример

стандартный ввод	стандартный вывод
4	YES
2 10 2 4	2 2

### Замечание

В примере из условия все числа входят в арифметическую прогрессию  $2, 4, 6, 8, 10, \dots$ , для которой  $s = 2$  и  $d = 2$ . Возможны и другие правильные ответы, например,  $s = 10$  и  $d = -2$ . Такой ответ тоже будет засчитан как верный.

## Задача Е. Маленькие знаменатели

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

Даны две дроби  $\frac{a}{b} < \frac{c}{d}$  с целыми неотрицательными числителями и целыми положительными знаменателями, а также выбрано некоторое целое положительное  $N$ . Рассмотрим все несократимые дроби  $\frac{e}{f}$  с целыми положительными числителями и знаменателями, не превосходящими  $N$ , лежащие между этими двумя дробями:  $\frac{a}{b} < \frac{e}{f} < \frac{c}{d}$ . Выпишем их в конечную последовательность в порядке возрастания сначала знаменателей, а потом числителей: дробь  $\frac{e_1}{f_1}$  будет в этом списке раньше  $\frac{e_2}{f_2}$ , либо если  $f_1 < f_2$ , либо если  $f_1 = f_2$  и  $e_1 < e_2$ . Ваша цель — вывести  $n$  первых дробей в этой последовательности (или всю последовательность, если в ней менее  $n$  дробей).

### Формат входных данных

В первой строке находятся шесть целых чисел  $a, b, c, d, N, n$  — две данные дроби, ограничение на числители и знаменатели и количество дробей, которые необходимо вывести ( $0 \leq a \leq 10^{18}$ ,  $1 \leq b, c, d, N \leq 10^{18}$ ,  $1 \leq n \leq 200\,000$ ). Гарантируется, что  $\frac{a}{b} < \frac{c}{d}$ , или, что равносильно,  $ad < bc$ .

### Формат выходных данных

В первой строке выведите целое число  $n'$  — количество дробей, которые вы выведете. В следующих  $n'$  строках выведите сами дроби в нужном порядке: в  $i$ -й строке должны быть целые  $e_i$  и  $f_i$ , разделённые пробелом — взаимно простые числитель и знаменатель  $i$ -й дроби ( $1 \leq e_i, f_i \leq N$ ). Число  $n'$  должно либо равняться  $n$ , либо лежать в пределах от 0 до  $n - 1$  и равняться количеству несократимых дробей между  $\frac{a}{b}$  и  $\frac{c}{d}$ , у которых числители и знаменатели целые положительные и не превосходят  $N$ .

### Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех групп с более строгими ограничениями, а также примеров.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	тесты из условия
1	6	$a, b, c, d, N \leq 100; n = 1$	0	—
2	12	$a, b, c, d, N \leq 1000; n \leq 20$	0, 1	—
3	15	$a, b, c, d \leq 2 \cdot 10^{12}; N \leq 200\,000$	0, 1, 2	—
4	30	$a, b, c, d, N \leq 10^9; n \leq 1000$	0, 1, 2	—
5	37	—	0, 1, 2, 3, 4	—

## Примеры

стандартный ввод	стандартный вывод
0 1 1 1 5 10	9 1 2 1 3 2 3 1 4 3 4 1 5 2 5 3 5 4 5
55 34 68 42 90 1	1 89 55
49 33 45 30 50 239	0

## Задача F. Произведение случайных чисел

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

Зафиксировано простое число  $p$ . Задана последовательность из  $n$  чисел, каждое от 1 до  $p - 1$ . Известно, что числа в последовательности выбраны случайно, равномерно и независимо друг от друга.

Выберите какие-то числа из последовательности так, чтобы их произведение, взятое по модулю  $p$ , оказалось равно заданному числу  $x$ . Если никакие числа не выбраны, произведение считается равным единице.

### Формат входных данных

В первой строке записаны три целых числа, разделённые пробелами: длина последовательности  $n$ , простое число  $p$  и искомое значение  $x$  ( $n = 100$ ,  $2 \leq p \leq 10^9$ ,  $0 < x < p$ ). Далее записаны  $n$  целых чисел, разделённых пробелами или переводами строк: последовательность  $a_1, a_2, \dots, a_n$  ( $0 < a_i < p$ ).

Гарантируется, что каждое число  $a_i$  выбрано случайно, независимо от остальных и равномерно из чисел  $1, 2, \dots, p - 1$ .

### Формат выходных данных

Выведите числа из последовательности, произведение которых по модулю  $p$  равно  $x$ . Эти числа должны быть выбраны из заданной последовательности. Формально: любое число должно встречаться в ответе не больше раз, чем оно встречается среди  $a_1, a_2, \dots, a_n$ . Порядок вывода чисел не важен. Разделяйте числа пробелами или переводами строк.

Если возможных ответов несколько, выведите любой из них.

### Система оценки

Каждый тест в этой задаче оценивается отдельно.

### Пример

стандартный ввод	стандартный вывод
100 11 4 9 6 1 1 10 4 9 10 3 1 10 1 6 8 3 3 9 8 10 3 7 7 1 3 3 1 5 2 10 4 1 5 6 7 2 6 2 8 3 3 6 7 6 3 1 5 10 2 2 10 9 6 8 6 2 10 3 2 7 4 3 2 8 6 4 1 7 2 10 8 4 9 7 9 8 7 4 7 3 2 8 2 3 7 1 5 2 10 7 1 8 6 4 10 10 3 6 10 2 1	4 6 10 9

### Замечание

Что делать в случае, когда ответа не существует? Поскольку вероятность генерации такого теста сильно меньше вероятности того, что ваше решение получит неправильный вердикт из-за воздействия на сервер лучей из космоса (или чего-нибудь ещё такого же удивительного), жюри считает, что отвечать на такие вопросы не требуется.

## Задача G. Отмерь и отрежь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Вам дана строка  $s$ . За одну операцию вы можете *отмерить любой* непустой собственный префикс  $p$  ( $0 < |p| < |s|$ ) строки  $s$ , совпадающий с суффиксом такой же длины, и *отрезать* либо **этот** префикс, либо **соответствующий** суффикс (но не оба сразу).

Например, в строке «abcab» есть лишь один префикс, совпадающий с соответствующим суффиксом — префикс «ab». Поэтому за одну операцию можно получить из неё либо «cab» (отрезав префикс), либо «abc» (отрезав суффикс). Никаких других строк за одну операцию получить нельзя.

Какой минимальной длины строки  $s$  можно добиться после применения оптимальной последовательности операций?

### Формат входных данных

На вход подаётся строка  $s$  ( $1 \leq |s| \leq 5000$ ), состоящая из маленьких английских букв.

### Формат выходных данных

Выведите минимально возможную длину строки после применения оптимальной последовательности операций.

### Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех групп с более строгими ограничениями, а также всех примеров.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0	—	—	тесты из условия
1	20	$n \leq 500$	0	—
2	40	$n \leq 2000$	0, 1	—
3	40	$n \leq 5000$	0, 1, 2	—

### Примеры

стандартный ввод	стандартный вывод
сааса	2
аабаа	2
abc	3

### Замечание

В первом примере возможная последовательность операций выглядит следующим образом:  $сааса \rightarrow са|аса \rightarrow аса \rightarrow ас|а \rightarrow ас$ .

В третьем примере невозможно применить ни одну операцию.

## Задача Н. Посольства

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

Есть  $n$  дружественных государств, в каждом из которых есть ровно по одному посольству каждого другого государства. Выведите общее количество посольств.

### Формат входных данных

В единственной строке находится целое число  $n$  ( $2 \leq n \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число — общее количество посольств.

### Система оценки

Каждый тест в этой задаче оценивается отдельно.

### Примеры

стандартный ввод	стандартный вывод
2	2
3	6

### Замечание

В примере с двумя государствами получится два посольства: посольство первого государства во втором и второго в первом.

## Задача I. Компоненты связности

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

Катя построила случайный неориентированный граф  $G$  из  $n = 1000$  вершин и  $m = 3000$  рёбер. Построение начинается с графа из  $n$  вершин без рёбер. Далее, пока рёбер не станет  $m$ , Катя псевдослучайным образом выбирает два целых числа  $u$  и  $v$  — независимо и равновероятно из всех целых чисел от 1 до  $n$ . Если  $u \neq v$  и рёбра между ними ещё нет, Катя добавляет его в граф.

Алиса хочет, чтобы Витя узнал количество компонент связности в этом графе. (Напомним, что две вершины лежат в одной компоненте связности, если из одной вершины в другую можно попасть, двигаясь по рёбрам графа.) Увы, она не может просто взять и передать ему это число. Вместо этого, когда Катя покажет свой граф, Алиса должна выбрать в нём ровно  $k = 20$  вершин:  $v_1, v_2, \dots, v_k$ . Тогда Катя пронумерует их в псевдослучайном порядке, то есть сопоставит числам  $v_i$  различные целые числа  $p(v_i)$  от 1 до  $k$  (все  $k!$  возможных сопоставлений имеют одинаковую вероятность). После этого Катя построит на них *индуцированный* граф  $G'$ : в нём будет ребро между вершинами  $p(u)$  и  $p(v)$ , если в исходном графе было ребро между соответствующими им вершинами  $u$  и  $v$ . После всего этого Катя пошлёт полученный граф Вите.

Помогите Алисе и Вите заранее договориться о том, как Алиса будет выбирать  $k$  вершин графа  $G$ , чтобы Витя, увидев граф  $G'$ , узнал точное количество компонент связности в графе  $G$ .

### Протокол взаимодействия

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

При первом запуске решение работает за Алису: получает случайный граф из  $n = 1000$  вершин и  $m = 3000$  рёбер и выбирает в нём  $k = 20$  вершин. В первой строке заданы числа  $n$  и  $m$  (они одинаковые во всех тестах). В следующих  $m$  строках заданы рёбра: каждая из них содержит номера двух вершин  $u$  и  $v$ , соединённых ребром ( $1 \leq u, v \leq n$ ,  $u \neq v$ , все рёбра различны). Гарантируется, что граф построен псевдослучайно, как описано в условии.

В первой строке выведете любые  $k = 20$  различных целых чисел от 1 до  $n$  в любом порядке: номера вершин, выбранных Алисой.

При втором запуске решение работает за Витю: получив граф  $G'$ , узнаёт количество компонент связности в исходном графе  $G$ . В первой строке заданы числа  $n'$  и  $m'$  — количество вершин и рёбер (здесь  $n' = k = 20$ ). В следующих  $m'$  строках заданы рёбра: каждая из них содержит номера двух вершин  $u'$  и  $v'$ , соединённых ребром ( $1 \leq u', v' \leq n'$ ,  $u' \neq v'$ , все рёбра различны). Гарантируется, что этот граф получен из исходного так, как описано в условии, используя ровно те вершины, которые были выбраны при первом запуске. Также гарантируется, что Катя нумерует вершины детерминированно: на каждом тесте при одинаковых выходных данных при первом запуске входные данные при втором запуске также будут одинаковы.

В первой строке выведете одно целое число — количество компонент связности в исходном графе.

### Система оценки

Каждый тест в этой задаче оценивается отдельно. Первые 10 тестов можно найти здесь: <https://acm.math.spbu.ru/olymp/2019-2020/>.

Баллы за них не начисляются, но они могут помочь при локальном тестировании. Кроме того, решения, которые не проходят первый тест, баллы не получают.

## 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.

стандартный ввод
1000 3000 607 728 995 31 ... (2997 других рёбер) 16 361
стандартный вывод
16 1 3 5 7 9 11 13 17 19 21 23 25 27 29 31 33 35 37 39
стандартный ввод
20 3 9 19 8 2 8 13
стандартный вывод
6

## Задача J. Конструктивное доказательство

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

Сочетанием из  $n$  элементов по  $k$  называется последовательность целых чисел  $a_1, a_2, \dots, a_k$ , в которой все числа лежат в пределах от 1 до  $n$  и записаны в порядке строгого возрастания. Число различных сочетаний из  $n$  элементов по  $k$  записывается как  $C_n^k$ . Например, при  $n = 4$  и  $k = 2$  существует шесть различных сочетаний: это  $(1, 2)$ ,  $(1, 3)$ ,  $(1, 4)$ ,  $(2, 3)$ ,  $(2, 4)$  и  $(3, 4)$ .

Широко известно следующее тождество:

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k.$$

Миша хочет увидеть *конструктивное доказательство* этого тождества: такое, которое каждому сочетанию из левой части формулы сопоставит сочетание из правой — либо из первого слагаемого, либо из второго. Напишите программу для Миши, которая каждому сочетанию из  $n$  элементов по  $k$  сопоставит одно из двух: либо слово «**first**» и сочетание из  $n - 1$  по  $k - 1$ , либо слово «**second**» и сочетание из  $n - 1$  по  $k$ . Затем, получив слово и сочетание, программа должна восстановить исходное сочетание из  $n$  элементов по  $k$ .

### Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. При вводе и выводе слова и числа в строках отделяются друг от друга пробелами.

При первом запуске решение по сочетанию из  $n$  элементов по  $k$  строит меньшее сочетание. В первой строке записано слово «**forward**». Вторая строка содержит целые числа  $n$  и  $k$  — параметры сочетания, за которыми следуют числа  $a_1, a_2, \dots, a_k$  — само сочетание ( $1 \leq k < n \leq 50$ ,  $1 \leq a_1 \leq a_2 \leq \dots \leq a_k \leq n$ ).

В первой строке выведите одно из двух: либо слово «**first**» и любое сочетание из  $n - 1$  по  $k - 1$ , либо слово «**second**» и любое сочетание из  $n - 1$  по  $k$ . Элементы сочетания записываются через пробел в порядке возрастания.

При втором запуске решение восстанавливает сочетание из  $n$  элементов по  $k$ . В первой строке записано слово «**reverse**». Вторая строка содержит целые числа  $n$  и  $k$  — параметры исходного сочетания, те же, что и при первом запуске, за которыми следуют либо слово «**first**» и сочетание из  $n - 1$  по  $k - 1$ , либо слово «**second**» и сочетание из  $n - 1$  по  $k$ . Слово и сочетание — ровно те, которые решение вывело при первом запуске.

В первой строке выведите восстановленное сочетание из  $n$  элементов по  $k$ : числа  $a_1, a_2, \dots, a_k$  в возрастающем порядке.

### Система оценки

Каждый тест в этой задаче оценивается отдельно.

## Примеры

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.

стандартный ввод	стандартный вывод
forward 4 2 1 2	first 1
reverse 4 2 first 1	1 2

Далее показаны два запуска какого-то решения на втором тесте.

стандартный ввод	стандартный вывод
forward 5 2 4 5	second 3 4
reverse 5 2 second 3 4	4 5