

## Задача А. Жадный царь

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В Либеральных Костромских Штатах правит жадный царь Норсу I. Недавно он решил, что в стране слишком много дорог, и хочет некоторые из них закрыть. Для этого он решил каждый день выполнять одну из следующих операций, пока это возможно:

- Если количество дорог больше или равно количеству городов, то царь выбирает любую из дорог и закрывает её.
- Если существует циклический маршрут, по которому можно, выехав из какого-то города, проехать по одной или более дорогам, и вернуться в тот же город, не проезжая ни по какой дороге дважды, то Норсу I выбирает любую из дорог этого маршрута, и закрывает её.
- Если существует город, до которого нельзя по дорогам добраться из столицы, то царь выбирает любую дорогу, которая выходит из этого города, и закрывает её.

Известно, что дороги в Либеральных Костромских Штатах двусторонние, всего в стране  $n$  городов, граф дорог является деревом, подвешенным за вершину с номером 1, которая является столицей, таким образом, что его рёбра идут из вершин с меньшими номерами в вершины с большими номерами, а также не существует ни одной дороги, ведущей из города в себя, и между любыми двумя городами проходит не более одной дороги.

Требуется определить, какая минимальная по количеству дорог дорожная сеть может получиться в результате таких действий.

### Формат входных данных

В единственной строке заданы  $n$  ( $1 \leq n \leq 1000$ ) чисел — для каждого города указан минимальный номер  $1 \leq a_i < i \leq n$  города с меньшим номером, с которым он соединён, если такой есть, или  $-1$ , если такого города нет.

### Формат выходных данных

Выведите в таком же формате, как и во вводе, описание получившейся дорожной сети на единственной строке. Если оптимальных решений несколько, разрешается выводить любое.

### Примеры

стандартный поток ввода	стандартный поток вывода
-1 1	-1 1

## Задача В. Страницы книги

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Библиотекарь занят починкой старинной книги. Книга — это текст известного автора на английском языке. К сожалению, переплёт рассыпался, и все страницы книги перемешались.

Библиотекарь отыскал точный текст книги в электронном виде. Кроме того, Библиотекарь отсканировал все страницы в порядке, в котором они сейчас лежат, и запустил программу распознавания текста. Однако книга старинная, поэтому распознать удалось не всё. Некоторые символы распознать невозможно, и в распознанном тексте вместо них стоит символ «#» (ASCII-код 35). Некоторые символы просто стёрлись, и в распознанном тексте вместо них стоит пробел. Наконец, некоторые символы стёрлись так, что в распознанном тексте они просто пропущены.

Помогите Библиотекарю. Для каждой страницы книги найдите её номер.

### Формат входных данных

Входные данные начинаются с текста книги, состоящего из нескольких строк. Далее идут распознанные тексты страниц, каждый из которых также занимает несколько строк. Каждая страница отделена от предыдущих входных данных специальной строкой вида «---###---###. .---###---»: 75 символов, состоящих из 25 чередующихся групп по три символа «-» и «#».

Текст книги — это электронная версия книги известного автора. Страницы же сгенерированы программой по следующему алгоритму.

Сначала книга разбивается на страницы так, чтобы каждая страница состояла из нескольких подряд идущих строк книги (в частности, каждая строка целиком попадает на какую-то одну страницу). Далее эти страницы случайным образом перемешиваются. В каждой книге не меньше трёх и не больше тысячи страниц. Каждая страница содержит от 30 до 80 строк включительно. Последняя страница книги может быть меньше, но всё же содержит не менее пяти строк.

После этого каждый непобельный символ на каждой странице независимо от других может быть распознан некорректно. А именно, с вероятностью 1% он заменён на символ «#», с вероятностью 1% заменён на пробел, с вероятностью 1% исчез, а с оставшейся вероятностью 97% оставлен без изменений. Пробелы и переводы строк всегда распознаются корректно. В исходном тексте книги встречаются только переводы строк и символы с ASCII-кодами от 32 до 126 включительно; символ «#» в нём не встречается.

Входные данные могут содержать пробелы в начале и в конце строк. Общий размер входных данных не превосходит семи мегабайт.

### Формат выходных данных

В первой строке выведите несколько целых чисел через пробел — номера страниц книги в том порядке, в котором они заданы во входных данных. Страницы должны быть пронумерованы целыми числами от единицы до их количества.

## Пример

стандартный поток ввода	стандартный поток вывода
A HAUNTED HOUSE	3 1 2
Whatever hour you woke there was a door shutting. From room to room they went, hand in hand, lifting here, opening there, making sure--a ghostly couple.	
"Here we left it," she said. And he added, "Oh, but here too!" "It's upstairs," she murmured. "And in the garden," he whispered. "Quietly," they said, "or we shall wake them."	
...	
Целиком тест можно увидеть в архиве	

## Замечание

Тесты в этой задаче — частично открытые. Чтобы тестировать решение локально было удобнее, можно скачать архив с примером и ещё девятью тестами: <https://ejudge.lksh.ru/olymp/book-pages.zip> (переводы строк для Windows) или <https://ejudge.lksh.ru/olymp/book-pages.tar.gz> (переводы строк для Linux). Эти тесты соответствуют первым десяти тестам в проверяющей системе в том же порядке. Последующие тесты держатся в секрете.

## Задача С. Правильная скобочная последовательность

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Как вы, возможно, знаете, множество  $\mathcal{S}$  правильных скобочных последовательностей из круглых и квадратных скобок можно рекурсивно определить следующим образом:

1.  $\varepsilon \in \mathcal{S}$  (пустая строка)
2.  $A \in \mathcal{S} \Rightarrow (A) \in \mathcal{S}$  (взятие в круглые скобки)
3.  $A \in \mathcal{S} \Rightarrow [A] \in \mathcal{S}$  (взятие в квадратные скобки)
4.  $A, B \in \mathcal{S} \Rightarrow AB \in \mathcal{S}$  (конкатенация)

Миша записал на доске правильную скобочную последовательность длины  $n$ , состоящую только из круглых скобок. После этого Петя подошёл к доске и заменил все скобки с  $l$ -й по  $r$ -ю включительно на соответствующие им квадратные скобки: «(» на «[» и «)» на «]». Как ни удивительно, после этого на доске оказалась вновь записана правильная скобочная последовательность.

Найдите любую последовательность, которая могла в итоге оказаться на доске, если гарантируется, что она существует.

### Формат входных данных

В первой строке записаны три целых числа  $n$ ,  $l$  и  $r$  ( $1 \leq l \leq r \leq n \leq 100$ ).

### Формат выходных данных

Выведите правильную скобочную последовательность, удовлетворяющую условию. Гарантируется, что такая последовательность существует. Если возможных ответов несколько, выведите любой из них.

### Примеры

стандартный поток ввода	стандартный поток вывода
4 2 3	( [ ] )
4 1 2	[ ] ( )

## Задача D. Подсчёт кирпичей

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Каменщик строит стену на вертикальной клетчатой плоскости, бесконечной во все стороны. Плоскость разделена на ряды высотой в одну клетку. Каждый ряд поделён на прямоугольники, состоящие из двух соседних клеток. Прямоугольники в соседних рядах сдвинуты на одну клетку.

Каждый прямоугольник плоскости либо пуст, либо содержит кирпич, заполняющий его целиком. Изначально некоторые прямоугольники уже содержат кирпичи. Каменщик может добавлять кирпичи в стену: если какие-то два соседних прямоугольника  $A$  и  $B$  в ряду содержат кирпичи, а прямоугольник  $C$ , лежащий сверху на их половинках, пуст, в прямоугольник  $C$  можно положить новый кирпич. Каменщик может добавлять кирпичи, пока получается найти такие три прямоугольника  $A$ ,  $B$  и  $C$ .

Помогите Каменщику. Выясните, какое максимальное количество кирпичей может получиться в стене после нуля или более добавлений.

### Формат входных данных

В первой строке записано целое число  $n$  — количество прямоугольников, содержащих кирпичи изначально ( $1 \leq n \leq 300\,000$ ). Каждая из следующих  $n$  строк содержит два целых числа  $row_i$  и  $col_i$  — строка и столбец левой клетки очередного прямоугольника, в котором изначально содержится кирпич ( $-10^9 \leq row_i, col_i \leq 10^9$ ,  $row_i$  и  $col_i$  имеют одинаковую чётность). Все заданные пары координат различны. Ряды нумеруются снизу вверх, а столбцы клеток — слева направо.

### Формат выходных данных

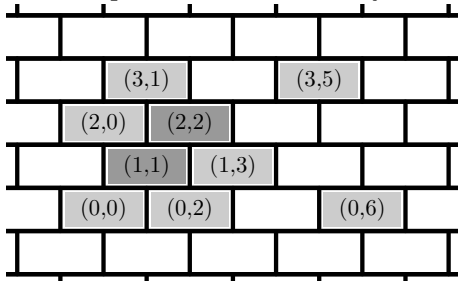
В первой строке выведите одно число — максимальное количество кирпичей в стене после нуля или более добавлений кирпичей.

### Пример

стандартный поток ввода	стандартный поток вывода
7 0 0 0 2 2 0 3 1 1 3 3 5 0 6	9

### Замечание

Иллюстрация для теста из условия:



## Задача Е. Фонари

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Снежок идёт по прямой дороге. Вдоль дороги в некоторых точках стоят фонари. У каждого фонаря есть координата и радиус освещения. Кроме того, каждый фонарь либо включён, либо выключен. Фонари пронумерованы в порядке возрастания координаты.

Если фонарь в координате  $x$  с радиусом освещения  $r$  включён, он освещает все точки дороги с координатами от  $x - r$  до  $x + r$  включительно. Если какая-то точка дороги не освещена ни одним фонарём, то в этой точке темно и опасно.

Снежок может перемещаться по дороге в любую сторону. Кроме того, если в той же точке, что и Снежок, находится фонарь, его можно включить или выключить. Изначально Снежок находится у первого фонаря, этот фонарь включён, а все остальные — выключены. Снежок хочет оказаться у последнего фонаря и при этом сделать так, чтобы последний фонарь был включён, а все остальные выключены. При этом Снежок не хотел бы ни в какой момент оказываться там, где темно и опасно.

Помогите Снежку. Найдите минимальную длину пути, который должен будет пройти Снежок, чтобы достичь цели, или выясните, что выполнить все условия невозможно.

### Формат входных данных

В первой строке записано целое число  $n$  — количество фонарей на дороге ( $2 \leq n \leq 100\,000$ ). Каждая из следующих  $n$  строк содержит два целых числа  $x_i$  и  $r_i$  — координату  $i$ -го фонаря на дороге и его радиус освещения ( $1 \leq x_i \leq 10^9$ ,  $1 \leq r_i \leq 10^9$ ). Фонари следуют в порядке строгого возрастания  $x_i$ .

### Формат выходных данных

В первой строке выведите одно число  $d$  — минимальную длину пути, который должен будет пройти Снежок, чтобы достичь цели, или  $-1$ , если выполнить все условия невозможно.

### Примеры

стандартный поток ввода	стандартный поток вывода	иллюстрация
5 1 5 3 1 4 9 7 8 8 4	21	
2 1 1 10 10	-1	
2 1 10 10 8	-1	
2 1 1000000000 1000000000 999999999	2999999997	

## Задача F. Множества Софи

Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

У Софи есть  $m$  множеств  $A_1, \dots, A_m$ , где каждое  $A_i$  может содержать только целые числа от 1 до  $n$ . Она очень обеспокоена разделяющей способностью этих множеств.

Будем говорить, что множество  $A$  отделяет  $X$  от  $Y$ , если  $(X \subset A)$  и  $(Y \cap A = \emptyset)$  или  $(Y \subset A)$  и  $(X \cap A = \emptyset)$ .

Назовём  $k$ -разделяющей способностью множеств  $A_1, \dots, A_m$  количество пар  $(X, Y)$ , для которых  $|X| = |Y| = k$ ,  $X \cap Y = \emptyset$  и существует такой индекс  $i$ , что  $A_i$  отделяет  $X$  от  $Y$ . Заметьте, что  $(X, Y)$  и  $(Y, X)$  считаются различными парами при  $X \neq Y$ .

Помогите Софи посчитать  $k$ -разделяющую способность её множеств. Так как ответ может быть очень большим, найдите его остаток по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке записаны три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq k \leq n \leq 100$ ,  $2k \leq n$ ,  $1 \leq m \leq 18$ ). Следующие  $m$  строк содержат описание множеств Софи. Если мы обозначим элементы  $A_i$  как  $\{a_i^{(1)}, \dots, a_i^{(|A_i|)}\}$ , то  $i$ -я из этих строк содержит разделённые пробелом целые числа  $|A_i|$ ,  $a_i^{(1)}$ ,  $\dots$ ,  $a_i^{(|A_i|)}$  в таком порядке.

### Формат выходных данных

Выведите единственное число —  $k$ -разделяющую способность множеств Софи по модулю  $10^9 + 7$ .

### Примеры

стандартный поток ввода	стандартный поток вывода
4 4 1 1 1 1 2 1 3 1 4	12
4 2 1 2 1 2 2 1 3	12
4 2 2 2 1 2 2 1 3	4

## Задача G. Юки 5x5

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Однажды два известных гения решили сыграть судьбоносную партию в Юки на частично заполненном поле. Игра проводится на клетчатом поле 5x5, игроки ходят **по очереди**, а в свой ход игрок **обязан** заполнить **одну** из еще незаполненных клеток. Если игрок должен сделать ход, а на поле есть полностью заполненная строка или полностью заполненный столбец или полностью заполненная главная диагональ или полностью заполненная побочная диагональ, этот игрок немедленно выигрывает. Определите победителя на данном поле при правильной игре.

### Формат входных данных

Вводится игровое поле в пяти строках по пять символов. Символ "." означает пустую клетку, символ **X** - занятую.

### Формат выходных данных

Выведите Kirei Kotomine, если выиграет первый игрок, иначе Kiritsugu Emiya.

### Пример

стандартный поток ввода	стандартный поток вывода
<pre>..XXX ....X XX... ..... ...X.</pre>	<pre>Kiritsugu Emiya</pre>

### Замечание

Заполненная главная диагональ выглядит так:

```
X....
.X...
..X..
...X.
....X
```

Заполненная побочная диагональ выглядит так:

```
....X
...X.
..X..
.X...
X....
```

## Задача Н. Гориллы в ЛКШ

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Наконец-то в ЛКШ стали принимать горилл! Правда из-за межвидовой дискриминации и устаревших консервативных предрассудков, гориллам разрешают заниматься только менеджментом посылок. Менеджмент посылок – это очень ответственная задача, требующая отвечать на следующие запросы:

1. Посылка  $id$  добавилась в конец очереди.
2. Посылка  $id$  добавилась в начало очереди.
3. Конечная посылка в очереди исчезла, если она существовала.
4. Начальная посылка в очереди исчезла, если она существовала.
5. Выведите  $id$  наиболее ранней посылки,  $id$  которой больше  $x$ , если она существует, иначе  $-1$ .
6. Выведите  $id$  наиболее поздней посылки,  $id$  которой больше  $x$ , если она существует, иначе  $-1$ .
7. Каждый  $id$  в очереди превращается в  $id \bmod x$ .
8. Каждый  $id$  в очереди превращается в  $id \& x$ .
9. Выведите сумму квадратов  $id$  в очереди по модулю  $2^{64}$ .

### Формат входных данных

В первой строке дано целое число  $n$  ( $1 \leq n \leq 2 \times 10^5$ ) - число запросов. В следующих  $n$  строках даны запросы. Сначала записано целое число  $type$  ( $1 \leq type \leq 9$ ) - тип запроса. Если  $type \in \{1, 2\}$ , то через пробел записано целое число  $id$  ( $1 \leq id \leq 10^9$ ). Если  $type \in \{5, 6, 7, 8\}$ , то через пробел записано целое число  $x$  ( $1 \leq id \leq 10^9$ ).

### Формат выходных данных

На каждый запрос, у которого  $type \in \{5, 6, 9\}$ , выведите ответ в отдельной строке.

### Пример

стандартный поток ввода	стандартный поток вывода
11	8
1 5	-1
2 7	-1
2 8	4
1 9	
5 5	
7 5	
6 4	
3	
4	
6 5	
9	

## Задача I. Вкусная шаверма

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В Ларьке Классной Шавермы готовят самую лучшую шаверму в городе. Там практикуют индивидуальный подход к каждому клиенту: клиенты могут заказать именно ту шаверму, которую хотят прямо сейчас. На случай, если клиенту очень понравилась заказанная им шаверма, но он боится забыть, как именно он просил её приготовить, вместе с каждым заказом выдаётся код, который позволяет восстановить, как готовилась соответствующая шаверма.

Этот код формируется следующим образом. Пусть всего доступно  $k \geq 2$  ингредиентов, пронумерованных от 0 до  $k - 1$ . Шаверма формируется последовательным добавлением нескольких ингредиентов. Ингредиенты могут повторяться. Если шаверма состоит из ингредиентов  $x_1, x_2, \dots, x_l$ , то она будет закодирована кодом  $x_1 \cdot k^{l-1} + x_2 \cdot k^{l-2} + \dots + x_{l-1} \cdot k + x_l$ . По техническим причинам, первый добавляемый ингредиент  $x_1 \neq 0$ . Понятно, что по коду можно однозначно восстановить последовательность ингредиентов, из которых была собрана шаверма.

Постоянный клиент этого ларька Иннокентий считает шаверму  $n$  похожей на шаверму  $m$ , если  $n$  получается из  $m$  выкидыванием некоторых ингредиентов с сохранением порядка оставшихся. Иннокентий помнит, что вчера он ел шаверму, имевшую код  $n$ , а сегодня — имеющую код  $m$ . Кроме того, он помнит, что вчерашняя шаверма была похожа на сегодняшнюю. Однако, он забыл, сколько всего бывает различных ингредиентов. Помогите ему восстановить какое-нибудь возможное значение  $k$ .

### Формат входных данных

В первой строке задан код вчерашней шавермы  $n$ . Во второй строке указан код сегодняшней шавермы  $m$  ( $1 \leq n \leq m \leq 10^{100}$ ).

### Формат выходных данных

В единственной строке выведите возможное количество ингредиентов  $k$ . Если ответов несколько, выведите любой из них. Если Иннокентий ошибся и ни одного возможного  $k$  не существует, выведите  $-1$ .

### Примеры

стандартный поток ввода	стандартный поток вывода
2	-1
3	-1
13	3
39	3

## Задача J. Племя шакалов

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вы, путешествуя по неизведанным уголкам нашей планеты, случайно встретили туземное племя Лающих Коварных Шакалов. Они схватили вас и привели в свою деревню. Все жители сбежались на вас посмотреть, и вы насчитали ровно  $n$  людей. Шаманы уверены в том, что вы — не человек, а дух, и собираются определить, добрый вы дух, или злой. В первом случае они осыплют вас своими дарами и отпустят, а во втором — зажарят на костре. Они жестами объяснили вам, как будет устроен ритуал, который определит их отношение к вам.

Жители деревни делятся на две касты: коварных шакалов, которые всегда лгут, и нековарных шакалов, которые всегда говорят правду. Ритуал пройдет следующим образом:  $m$  различных жителей по очереди подойдут к вам, и каждый скажет, сколько по его мнению в деревне нековарных шакалов. Ни про какого жителя вам не будет сообщено, к какой касте он относится. А вы после этого должны будете сказать, каково максимально возможное количество нековарных шакалов в деревне, не противоречащее всей имеющейся у вас информации. Вас пугает вид разжигаемого туземцами костра, а потому вы хотите ответить на заданный вопрос правильно.

Возможно, вам все это снится в бредовом кошмаре, и описываемой ситуации происходить не может.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$ , общее число жителей в деревне и количество высказавшихся, соответственно ( $1 \leq n \leq 10^{18}, 0 \leq m \leq 10^5, m \leq n$ ). Далее в  $m$  строках идет по одному числу  $a_i$  — сколько нековарных шакалов в деревне по мнению высказавшегося под номером  $i$  ( $-10^{18} \leq a_i \leq 10^{18}$ ).

### Формат выходных данных

Выведите одно число — максимально возможное количество нековарных шакалов среди всех жителей деревни. Если описываемая ситуация невозможна, и вам срочно нужно проснуться, выведите одно число -1.

### Примеры

стандартный поток ввода	стандартный поток вывода
5 3 1 2 3	3

### Замечание

В примере из условия первые два высказавшихся могут быть коварными шакалами, а третий — нековарным, если при этом два оставшихся жителя тоже нековарные, то всего их получится три.

## Задача К. ABC-кодирование

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В Лаборатории Компьютерного Шаловства недавно разработали уникальный способ сжатия данных — ABC-кодирование. Его Вам сейчас и предстоит запрограммировать.

Для сжатия строки  $s$  предлагается найти две строки  $A$  и  $B$ , и представить  $s$  в виде конкатенации нескольких копий  $A$  и  $B$ . Например, для строки «abacaba» можно выбрать  $A = \text{«aba»}$ ,  $B = \text{«c»}$  и выписать код  $C = ABA$ . Длиной кода будет  $|A| + |B| + |C|$ . В данном случае мы получили  $|A| = 3$ ,  $|B| = 1$ ,  $|C| = 3$  и длину ABC-кода 7.

Для простоты кодирования на выбор  $a$  и  $b$  наложенные следующие ограничения: они не могут быть пустыми и не могут являться префиксами друг друга. Например, для строки «xxxxxx» нельзя выбрать  $A = \text{«xxx»}$ ,  $B$  пустая, и выписать код  $AA$ . Взять  $A = \text{«xxx»}$ ,  $B = \text{«x»}$ . Зато можно взять  $A = \text{«xxx»}$ ,  $B = \text{«y»}$ , и выписать код  $AA$ .

Задача — по данной строке построить ABC-код минимальной длины.

### Формат входных данных

Строка  $s$  длины от 2 до 2000, состоящая только из символов 'a' и 'b'.

### Формат выходных данных

На первой строке выведите длину оптимального ABC-кода.

Далее выведите  $a$ ,  $b$ ,  $c$  по одной в строке.

Если оптимальных ответов несколько выведите любой.

### Примеры

стандартный поток ввода	стандартный поток вывода
ababbbbbbbababbbb	12 ab bbb AABBAAB