

## Задача А. Фальшивые монеты

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

В королевстве Байтландия завелись гоблины-фальшивомонетчики, выпускаемые ими монеты исчезают на следующее утро.

По дороге домой гном Гектар нашел мешочек с  $n$  монетками и на радостях истратил  $m$  из них в магазине. Проснувшись с утра, Гектар с огорчением заметил, что у него осталось всего  $k$  монеток. Гектар очень добропорядочный гном, поэтому он хочет сообщить об этом происшествии в эльфийскую полицию. Для этого ему необходимо знать, сколько фальшивых монет он нашел.

Помогите Гектару определить минимальное и максимальное количество фальшивых монет, которые могли быть в мешочке.

### Формат входных данных

Первая строка содержит число  $p$  ( $1 \leq p \leq 10^4$ ) — количество наборов тестов. Далее следует  $p$  строк наборов.

Первая строка каждого набора содержит три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq m \leq n \leq 100$ ,  $0 \leq k \leq n - m$ ) — число монеток в мешочке, число монеток, потраченных в магазине и число монет, оставшихся у гнома с утра. Гарантируется, что входные данные корректны.

### Формат выходных данных

Для каждого набора выведите два целых числа — минимальное и максимальное количество фальшивых монет, которые могли быть в мешочке.

### Примеры

стандартный ввод	стандартный вывод
2	1 3
5 2 2	0 3
8 3 5	
3	4 5
5 1 0	2 3
5 1 2	0 1
5 1 4	

## Задача В. Зашифрованное сообщение

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

Все королевские письма в Байтландии шифруются специальным устройством для защиты от шпионов. Для проверки, есть ли ошибка в зашифрованном сообщении, используются различные методы, самым простым из которых является сравнение числа единиц и нулей в сообщении.

Помогите королевским шифровальщикам определить, какие символы преобладают в сообщении: нули или единицы.

### Формат входных данных

Единственная строка содержит непустую строку, состоящую из нулей и единиц. Длина строки не превышает 10.

### Формат выходных данных

В единственной строке выведите «EQUAL», если строка содержит равное число нулей и единиц, «ZERO», если в строке больше нулей и «ONE» в противном случае.

### Примеры

стандартный ввод	стандартный вывод
0001101	ZERO
10	EQUAL
111	ONE

## Задача С. Сокровище

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

Бен Ганн нашел в пещере сокровища Флинта. Так как Флинт был очень хитрым пиратом, то он поместил его в один из многочисленных сундуков, стоящих в ряд и пронумерованных по порядку от 1 до  $n$ . На каждом сундуке есть одна из трех надписей: сокровища лежат в сундуке с большим номером, сокровища лежат в сундуке с меньшим номером, сокровища лежат в этом сундуке. Флинт был еще и безграмотным пиратом, поэтому надписи могут не соответствовать действительности.

Бен Ганн быстро отыскал сокровища и решил поместить их в один из сундуков и исправить некоторые надписи так, чтобы все они соответствовали действительности. Бен Ганн решил выбрать такой сундук, чтобы количество надписей, которые надо исправить, было минимально. Помогите Бену Ганну найти минимальное количество исправлений, которые ему придется сделать.

### Формат входных данных

Входные данные содержат несколько тестовых примеров. Первая строка содержит число  $T$  — число тестовых наборов ( $1 \leq T \leq 100$ ). Далее идут тестовые наборы в следующем формате.

Первая строка каждого набора содержит одно натуральное число  $n$  — количество сундуков. Вторая строка содержит описание надписей на сундуках. Если  $i$ -е число равно  $-1$  или  $1$ , это означает что на сундуке номер  $i$  написано, что сокровища лежат в сундуке с номером, меньше или больше, чем  $i$ , соответственно. Если  $i$ -е число равно  $0$ , это означает, что на  $i$ -м сундуке написано, что сокровища лежат в нем.

Суммарное число всех сундуков в одних входных данных не превосходит  $10^5$ .

### Формат выходных данных

Для каждого тестового набора в порядке появления во входных данных выведите одно число — минимальное число надписей, которое нужно исправить Бену.

### Пример

стандартный ввод	стандартный вывод
2	4
5	0
-1 -1 0 1 1	
5	
1 1 0 -1 -1	

## Задача D. Простой SAT

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

В университете Вася недавно изучил проблему разрешимости булевых формул — по заданной булевой формуле найти такой набор значений, при котором эта формула истинна. Он расстроился, узнав, что не существует полиномиального алгоритма решения данной задачи. В связи с этим Вася решил модифицировать задачу следующим образом: заданный набор булевых значений, не меняя их порядок, соединить операторами «and» и «or» так, чтобы получившаяся формула принимала истинное значение. Среди всех таких наборов операторов Вася хочет найти тот, в котором количество операторов «or» минимально.

Помогите Васе решить такую задачу. Напомним, что «and» возвращает истину тогда и только тогда, когда оба аргумента истинны, а «or» возвращает истину, когда хотя бы один из аргументов истина. Приоритет оператора «and» выше, чем приоритет оператора «or».

### Формат входных данных

Первая строка содержит одно натуральное число  $k$  — количество наборов входных данных. Следующие  $2k$  строк содержат описания самих наборов.

Первая строка описания очередного набора содержит одно целое число  $n$  ( $2 \leq n \leq 10^4$ ) — количество булевых значений.

Вторая строка, состоящая из нулей и единиц, содержит  $n$  символов — последовательность булевых значений. Единица обозначает истину, ноль обозначает ложь.

Гарантируется, что суммарное количество булевых значений во всех наборах входных данных одного теста не превышает  $10^4$ .

### Формат выходных данных

Для каждого набора входных данных выведите одну строку — ответ на задачу.

В случае, если ответ для данного набора существует, выведите  $n - 1$  слово.  $i$ -е слово должно быть равно «and» в случае, если между  $i$ -м и  $i + 1$ -м значениями должна стоять конъюнкция, и «or» в противном случае.

Если не существует способа расставить операторы так, чтобы формула принимала истинное значение, выведите одну строку, содержащую «Impossible»

### Пример

стандартный ввод	стандартный вывод
2	and or
3	Impossible
011	
2	
00	

## Задача E. Проверка SAT

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

Вася написал решение к задаче «Простой SAT»: задан набор булевых значений, и, не меняя их порядок, необходимо соединить их операторами «and» и «or» так, чтобы получившаяся формула принимала истинное значение, а количество операций «or» было минимальным. Друг Васи Петя заинтересовался этой задачей и тоже написал решение. Вася не доверяет Пете, поэтому решил написать программу, которая сравнит их решения.

Программа Васи должна возвращать три различных результата:

1. «WA», если решение Васи нашло корректный ответ, а решение Пети не нашло ответ и вывело «Impossible», или содержит больше операций «or», чем ответ Васи;
2. «Fail», если решение Пети нашло корректный ответ, а решение Васи не нашло ответ и вывело «Impossible», или содержит больше операций «or», чем ответ Пети;
3. «OK», если решение Пети корректно, и количество «or» совпадает с количеством «or» в решении Васи или оба решения вывели «Impossible».

Помогите Васе написать такую программу. Напомним, что «and» возвращает истину тогда и только тогда, когда оба аргумента истинны, а «or» возвращает истину, когда хотя бы один из аргументов истинен. Приоритет оператора «and» выше, чем приоритет оператора «or».

### Формат входных данных

Первая строка содержит одно целое число  $k$  — количество наборов входных данных, которые необходимо обработать. Далее следуют  $k$  описаний самих наборов, каждое из которых состоит из четырех строк.

Первая строка содержит одно целое число  $n$  — количество булевых значений.

Вторая строка, состоящая из нулей и единиц, содержит  $n$  символов — последовательность булевых значений. Единица обозначает истину, ноль обозначает ложь.

Третья строка, состоящая из «or» и «and», содержит  $n - 1$  слово, или слово «Impossible» — решение Васи.

Четвертая строка содержит решение Пети в том же формате.

Гарантируется, что сумма  $n$  во всех наборах входных данных в одном тесте не превышает  $10^5$ .

### Формат выходных данных

Для каждого набора входных данных выведите «WA», «Fail», или «OK».

## Пример

стандартный ввод	стандартный вывод
4	OK
4	WA
1001	Fail
or and and	Fail
and and or	
3	
000	
Impossible	
or and	
3	
100	
Impossible	
or and	
3	
111	
or or	
or and	

## Задача F. Подстрока

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

В этой задаче требуется найти максимальную по длине подстроку данной строки такую, что каждый символ встречается в ней не более  $k$  раз.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $1 \leq k \leq n$ ), где  $n$  — количество символов в строке.

Вторая строка содержит  $n$  символов — данную строку, состоящую только из строчных латинских букв.

### Формат выходных данных

Выведите два числа — длину искомой подстроки и номер ее первого символа. Символы нумеруются с единицы. Если решений несколько, выведите любое.

### Примеры

стандартный ввод	стандартный вывод
3 1 abb	2 1
5 2 ababa	4 1

## Задача G. Родственные геномы

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

Геном представляет собой последовательность генов, где каждый ген закодирован каким-то числом. Гены в геноме пронумерованы последовательно, начиная с единицы. В лаборатории по изучению генома учеными было сделано открытие, что два генома являются родственными, если они имеют одинаковое количество генов и один геном можно получить из другого последовательностью из не более чем  $k$  следующих операций:

- Поменять первую и вторую половину генома местами.
- Поменять гены, стоящие на четных местах, с генами, стоящими на нечетных местах,  $2i$ -й ген меняется местами с  $2i + 1$ -м.

Вам даны два генома одинакового размера. Определите, являются они родственными или нет.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  — размер геномов и число максимальное шагов ( $2 \leq n \leq 100$ ;  $0 \leq k \leq 100$ ),  $n$  — четно.

Вторая и третья строка входных данных содержат по  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — первый и второй геномы, соответственно.

### Формат выходных данных

Если два генома являются родственными в первой строке выведите число  $m$  — число действий, необходимых, чтобы получить второй геном из первого.

В следующей строке выведите последовательность из  $m$  чисел 1 или 2 — номера операций, которые надо применить для получения второго генома из первого. Если существует несколько решений, выведите любое из них.

Если геномы не являются родственными, в первой и единственной строке выведите «Impossible».

### Примеры

стандартный ввод	стандартный вывод
4 1 1 2 3 4 3 4 1 2	1 1
4 1 1 2 3 4 2 1 4 3	1 2
4 2 1 2 3 4 4 3 2 1	2 1 2
4 1 1 2 3 4 4 3 2 1	Impossible

## Задача N. Генератор перестановок

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

Коля недавно изучил алгоритм Фишера-Йетса, который позволяет написать программу генерирующую случайную перестановку первых  $n$  натуральных чисел. А именно, эта программа может быть устроена следующим образом:

```
var a = Array<Int>(n, {i -> i + 1})  
for (i in 1..n) {  
    val t = random.nextInt(i) + 1  
    val tmp = a[i - 1]  
    a[i - 1] = a[t]  
    a[t] = tmp  
}
```

Здесь `random.nextInt(z)` возвращает случайное число в пределах от 0 до  $z - 1$ .

После нескольких экспериментов с этим алгоритмом, Коля немного изменил третью строку и в результате получился такой алгоритм:

```
var a = Array<Int>(n, {i -> i + 1})  
for (i in 1..n) {  
    val t = random.nextInt(Math.min(i, k)) + 1  
    val tmp = a[i - 1]  
    a[i - 1] = a[t]  
    a[t] = tmp  
}
```

Здесь  $k$  является некоторой константой. Коля заметил, что получившийся алгоритм может генерировать не любую перестановку. Множество перестановок, которые он может сгенерировать, зависит от значения  $k$ . Ему стало интересно, при каком минимальном значении константы  $k$ , в результате его исполнения может получиться заданная перестановка  $P$ .

Помогите Коле выяснить минимальное значение  $k$ , при котором может быть сгенерированна заданная перестановка.

### Формат входных данных

Первая строка содержит число  $t$  — количество входных наборов в тесте. Далее следует  $2t$  строк, описывающих наборы.

Первая строка каждого описания содержит одно число  $n$  ( $1 \leq n \leq 10^5$ ) — размер перестановки. Вторая строка содержит заданную перестановку  $P$  чисел от 1 до  $n$ . Числа разделены пробелами.

Гарантируется, что сумма  $n$  по всем входным наборам в одном тесте не превосходит  $3 \cdot 10^5$

### Формат выходных данных

Для каждого входного набора на отдельной строке выведите искомое минимальное значение  $k$ .

### Пример

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

## Задача I. Созвездие

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Астрономы нашли на небе новую туманность «Серебристый путь», представляющую собой набор звезд, некоторые из которых соединены между собой газовыми путями. В связи с законами физики, газовые пути не могут образовывать цикл, а любые две звезды соединены между собой газовыми путями, возможно через другие звезды.

Теперь ученые хотят выделить самое красивое созвездие типа «Звезда» внутри туманности. Созвездием типа «Звезда» называется такой набор звезд, где существует центральная звезда и лучи одинаковой длины, исходящие из нее. Лучом называется набор звезд, соединенных по цепочке, друг с другом, газовыми путями. Длиной луча называется число газовых путей в нем. Никакая звезда, кроме центральной, не может находиться одновременно в двух лучах. Среди всех таких созвездий астрономы хотят выбрать самое красивое. Красота созвездия определяется как произведение длины лучей на количество лучей в нем.

Помогите ученым найти самое красивое созвездие.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $2 \leq n \leq 10^5$ ) — число звезд.

В следующих  $n-1$  строках находится описание газовых путей:  $i$ -й газовый путь задается числами  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ) — номерами звезд, соединяемых  $i$ -м газовым путем.

### Формат выходных данных

В единственной строке выведите три числа: номер центральной звезды самого красивого созвездия, количество лучей в нем и длину лучей.

Если самых красивых созвездий несколько, выведите описание любого из них.

### Пример

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

## Задача J. Лабиринт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вы участвуете в соревновании роботов компании Jet Vots. Оно заключается в следующем. Вы контролируете  $k$  роботов, которые находятся в одном лабиринте. Лабиринт представляет собой прямоугольное клетчатое поле, в котором каждая клетка или свободна или занята стенкой.

Для каждого робота заданы начальная и конечная клетки. Каждую секунду каждый робот получает команду сдвинуться в каком-то из четырёх направлений: вверх, вниз, влево или вправо. Команды могут быть разными для разных роботов. Если соседней клетки в указанном направлении нет или там находится стенка есть стенка, то робот никуда не двигается, иначе он перемещается в соседнюю клетку в соответствующем направлении. В каждой клетке может находиться одновременно сколько угодно роботов.

Определите, какое минимальное время потребуется для того, чтобы все роботы одновременно оказались в конечных позициях.

### Формат входных данных

Первая строка содержит два целых числа  $n, m$  ( $1 \leq n, m \leq 100$ ) — высота и ширина лабиринта, соответственно.

Следующие  $n$  строк содержат по  $m$  символов «.» или «#». «.» означает, что в данном месте лабиринта нет стенки, а «#» означает, что в данном месте стенка.

$n + 2$ -я строка содержит целое число  $k$  ( $1 \leq k \leq 100$ ) — количество подконтрольных роботов. Каждая из последующих  $k$  строк содержит четыре числа  $x_s, y_s, x_f$  и  $y_f$  — координаты начальной и конечной клетки, соответственно.  $x_s$  и  $x_f$  соответствуют строкам,  $y_s$  и  $y_f$  — столбцам. Строки пронумерованы сверху вниз от 1 до  $n$ , столбцы — слева направо от 1 до  $m$ . В начальной клетке каждого робота стенок нет. Роботы могут иметь совпадающие начальные или конечные позиции.

### Формат выходных данных

Выведите минимальное время, за которое всех роботов можно одновременно поставить в конечные позиции, или «Impossible», если это невозможно сделать.

### Примеры

стандартный ввод	стандартный вывод
3 2 .# .. .. 2 1 1 2 2 2 2 1 1	2
2 3 .#. #.. 2 1 1 2 3 2 3 1 1	Impossible