

## Задача А. Домашняя акула

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

Недавно Леша завел себе домашнюю акулу. Акула — достаточно сложный домашний питомец в плане содержания, хотя бы потому, что за один раз она может съесть только **ровно**  $k$  единиц **однотипного** корма.

Леша решил показать акулу своему другу Диме, но тот испугался ее и предложил Леше следующий спор. Дима принесет Леше некоторый корм, а Леша попытается скормить его акуле каким-либо образом (возможно, в несколько приемов). Если Леше не удастся этого сделать, он проиграл.

Дима разложил  $n$  единиц имеющегося у него корма в ряд на столе так, что тип  $i$ -й единицы корма для акул на столе был равен  $a_i$ . Леша не знает, принесет ли Дима весь свой корм, но уверен, что Дима принесет последовательный интервал того, что у него сейчас на столе. Найдите количество интервалов корма таких, что Леша победит в заключенном пари!

### Формат входных данных

В первой строке входного файла содержатся два целых числа  $n$  и  $k$  — количество единиц корма на столе у Димы и количество единиц корма, которое акула может съесть за один раз.

Во второй строке содержатся  $n$  разделенных пробелом целых чисел  $a_1, a_2, \dots, a_n$ , где  $a_i$  — тип  $i$ -й единицы корма.

$$1 \leq n, k \leq 2 \cdot 10^5$$

$$1 \leq a_i \leq n$$

### Формат выходных данных

Выведите одно целое число — количество последовательных интервалов корма, для которых Леша победит в споре с Димой.

### Примеры

стандартный ввод	стандартный вывод
6 2 1 2 1 2 1 2	3
6 6 1 1 1 1 1 1	1
9 3 1 2 3 1 2 3 1 2 3	1

## Задача В. Перестановку на прокачку

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

Вы наверняка знаете, что такое перестановка, но на всякий случай, напомним:

Перестановкой из  $n$  элементов называется последовательность  $p_1, p_2, \dots, p_n$ , в которой все числа от 1 до  $n$  встречаются один раз.

Для последовательности  $a_1, a_2, \dots, a_n$  также определим количество инверсий как количество пар  $i, j$  таких, что  $i < j$  и  $a_i > a_j$ . Красотой последовательности длины  $m$  назовем число  $m$  минус количество инверсий в этой последовательности.

Вам дана перестановка из  $n$  элементов. Разрешено вычеркнуть из нее любое количество чисел. Найдите, какую максимальную красоту последовательности можно получить.

### Формат входных данных

В первой строке входного файла содержится одно целое число  $n$  — количество элементов в перестановке.

Во второй строке содержатся  $n$  разделенных пробелом целых чисел  $p_1, p_2, \dots, p_n$  — исходная перестановка.

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

$$1 \leq p_i \leq n$$

Все  $p_i$  различны.

### Формат выходных данных

Выведите одно целое число — значение максимальной красоты, которую можно получить.

### Примеры

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

### Замечание

В первом тестовом примере подойдет последовательность «1 2 3 4» или «1 2 3 5», обе с красотой 4.

Во втором тестовом примере можно взять всю последовательность «2 1 3 4 5 6» с красотой 5.

## Задача C. $A + B = C$

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

Вам дано три целых числа. Вам требуется упорядочить их так, чтобы сумма первых двух была равна третьему или определить, что такого порядка не существует.

### Формат входных данных

В первой строке задается три целых числа  $a_i$ .

$$-2^{63} < a_i < 2^{63}$$

### Формат выходных данных

Выведите три заданных числа в искомом порядке. Если такого порядка не существует, то выведите "-1 -1 -1". Если существует несколько решений, то выведите любой из них.

### Примеры

стандартный ввод	стандартный вывод
2 1 3	1 2 3
2 4 42	-1 -1 -1

## Задача D. Петя и массив

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

У Пети есть массив состоящий из  $n$  чисел и любимое число  $k$ .

Для любого массива Петя умеет определять его привлекательность по формуле:  $\max(array) - \min(array)$ , где  $\max(array)$  обозначает максимальный элемент в массиве, а  $\min(array)$  — минимальный элемент массива.

Для того, чтобы массив приблизился по красоте к его любимому числу, Петя может удалить **не более одного элемента** из этого массива. В случае если он может получить массив с привлекательностью не больше числа  $k$ , то он называет изначальный массив — красивым.

Петя хочет, чтобы вы ответили на  $q$  его вопросов. Каждый вопрос состоит из двух чисел  $L$  и  $R$ . Ему интересно сколько различных красивых подотрезков массива лежит внутри данного отрезка от  $L$  до  $R$ . Более формально, вам нужно найти количество различных пар чисел « $l$   $r$ » для которых выполняется:  $L \leq l < r \leq R$  и массив  $a_l, a_{l+1}, \dots, a_r$  является красивым.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $q$  и  $k$  — количество чисел в массиве, количество вопросов и любимое число Пети.

Во второй строке содержится  $n$  чисел  $a_1, a_2, \dots, a_n$  — описание массива Пети.

Следующие  $q$  строк содержат по два целых числа  $L$  и  $R$  — описание очередного вопроса Пети.

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

$$0 \leq a_i, k \leq 10^9$$

$$1 \leq L \leq R \leq n$$

### Формат выходных данных

Вы должны вывести для каждого вопроса Пети одно целое число в отдельной строке — количество красивых подотрезков длины больше одного для описанного отрезка.

### Примеры

стандартный ввод	стандартный вывод
5 1 4 0 10 10 2 4 1 5	7
5 1 4 0 10 1 2 4 1 5	10

## Задача Е. Автостопом по галактике

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

Недавно Леша нашел в библиотеке старинную книгу: «Путеводитель для путешествующих автостопом по галактике». В ней рассказывалось, что всего в галактике  $n$  звезд, пронумерованных от 0 до  $n - 1$ , и для каждой звезды  $i$  известно  $f_i$  — звезда, на которую вас готовы подвезти от звезды  $i$ .

Леша захотел оценить насколько данный вид путешествий по галактике эффективен. Для этого он посчитал  $c_i$  — количество различных звезд, которые можно посетить, начав путешествие со звезды  $i$ . Тогда привлекательностью автостопа назовем сумму  $c_i$  по всем звездам.

Книга была настолько старой, что некоторые числа в ней оказались затерты. У вас есть прекрасная возможность доказать Леше, что автостоп — лучший способ путешествий по галактике: допишите затертые  $f_i$  самостоятельно так, чтобы привлекательность автостопа была максимальна.

### Формат входных данных

В первой строке входного файла содержится одно целое число  $t$  — количество тестов. Каждые следующие две последовательные строки описывают один тест, причем в первой строке содержится одно целое число  $n$ . Во второй строке содержатся  $n$  разделенных пробелом целых чисел  $f_0, f_1, \dots, f_{n-1}$ .  $f_i = -1$  означает, что звезда, куда вас могут подвезти от звезды  $i$  неизвестна.

$$1 \leq t, n \leq 5 \cdot 10^5$$

$$-1 \leq f_i \leq n - 1$$

Сумма  $n$  по всем тестам не превосходит  $5 \cdot 10^5$ .

### Формат выходных данных

Для каждого теста выведите одну строку содержащую одно число: максимальную привлекательность автостопа.

### Пример

стандартный ввод	стандартный вывод
2	3
5	42
0 1 2 -1 -1	
7	
-1 -1 -1 -1 -1 -1 -1	

## Задача F. Чемпионат по футболу

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

Леша, Дима и Влад поспорили, кто из них лучше всего играет в футбол. Для того, чтобы провести между собой чемпионат, они позвали  $n$  своих друзей.

Осталось определиться, в каких составах будет проходить чемпионат. Друзья получили футболки с номерами от 1 до  $n$  так, что про друга с  $i$ -м номером известно, что его скилл в футболе численно равен  $i$ . Тогда сила команды это сумма сил отдельно взятых игроков.

Справедливости ради, разбиться на команды было решено так, чтобы все команды были равны по силе, ведь ребят интересует сравнить силы между собой. Чтобы не обидеть ни одного из друзей, каждый из них должен попасть ровно в одну команду. Строгих правил на количество игроков в команде нет — каждая команда может состоять из произвольного количества игроков.

### Формат входных данных

В первой строке входного файла содержится одно целое число  $n$  — количество друзей.

$$1 \leq n \leq 10^6$$

### Формат выходных данных

Если разбить на три равносильные команды невозможно, выведите «Impossible».

Иначе, выведите «Possible» и 6 строк с описанием 3-х получившихся команд. В первой строке описания команды выведите  $k$  — количество игроков набранных в эту команду. Во второй строке выведите  $k$  чисел — номера на футболках друзей в этой команде.

### Примеры

стандартный ввод	стандартный вывод
6	Possible 2 6 1 2 5 2 2 4 3
9	Possible 2 9 6 2 8 7 5 5 4 3 2 1
10	Impossible

## Задача G. Раскраски

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

Мальчику Вове очень нравятся раскраски. На этот раз, Вова решил использовать в качестве раскраски свою тетрадь в клеточку. Каждый лист имеет высоту в  $n$  клеточек и ширину в  $m$  клеточек. У Вовы есть фломастер, с помощью которого он и собирается раскрашивать листы тетради. Чтобы процесс раскраски был более интересным, Вова решил установить определенные правила:

- Каждая клетка должна быть либо полностью закрашена, либо не закрашена вовсе;
- Если Вова раскрасил клеточку  $(x, y)$ , то следующая закрашиваемая клетка должна быть соседней по стороне;
- Начинать раскрашивать можно с любой клетки;
- Можно закрасить ранее закрашенную клеточку, в таком случае ничего не произойдет, так как клетка уже была закрашена.

Вова хочет нарисовать все возможные варианты рисунков, которые можно получить, раскрашивая листы тетрадки по таким правилам. Вова интересуется, а сколько листов ему для этого понадобится (листы односторонние). Два рисунка будут считаться различными, если найдется такая клеточка  $(x, y)$ , что в одном рисунке она будет закрашена, а в другом нет.

### Формат входных данных

В единственной строке задано два целых числа  $n$  и  $m$  — высота и ширина листа тетради в клеточках.

$$1 \leq n, m \leq 12$$

### Формат выходных данных

Выведите единственное целое число — количество листов, которые Вове понадобятся, чтобы получить все возможные рисунки.

Так как ответ может быть слишком большим, его необходимо вывести по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
2 2	13
3 3	218

## Задача Н. Полицейский с Рублевки

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

В отделе полиции в Барвихе работают не самые профессиональные сотрудники. Так, для получения премии за 2018 год, было решено немного исправить полицейский отчет.

Полицейский отчет — это последовательность  $a_1, a_2, \dots, a_n$ , где  $a_i$  — сложность  $i$ -го раскрытого преступления. Для корректности работы программы все числа в отчете должны быть **целыми**.

Полицейские отчеты хранятся в программе «Vxcel». Программа имеет следующую защиту: при попытке подменить отчет за прошедшие годы, программа не сверяет измененный файл со старым полностью, а сверяет лишь некоторые статистические данные.

Два файла считаются программой одинаковыми, если одновременно совпадают среднее арифметическое чисел в этих файлах, а так же дисперсия этих чисел. То есть, если в первом файле среднее равно  $m_1$ , дисперсия равна  $D_1$ , а во втором  $m_2$  и  $D_2$  соответственно, то два файла считаются одинаковыми, если  $m_1 = m_2$  и  $D_1 = D_2$ .

Дисперсия  $D$  в «Vxcel» считается следующим образом ( $m$  — среднее арифметическое):

$$D = \frac{1}{n} \sum_{i=1}^n (a_i - m)^2$$

Среднее арифметическое в свою очередь считается по формуле:

$$m = \frac{1}{n} \sum_{i=1}^n a_i$$

Обратите внимание, что дисперсия и среднее арифметическое могут являться не целыми числами.

Полицейский Леша решил убрать одно из раскрытых преступлений, а взамен дописать два вымышленных. Найдите любой способ сделать это так, чтобы «Vxcel» не обнаружила подмены.

### Формат входных данных

В первой строке входного файла содержится одно целое число  $n$  — количество раскрытых преступлений.

Во второй строке содержатся  $n$  разделенных пробелом целых чисел  $a_1, a_2, \dots, a_n$  — сложности раскрытых преступлений.

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

### Формат выходных данных

В первой строке выведите «Impossible», если такая подмена невозможна.

Иначе, в первой строке выведите «Possible», а во второй три целых числа  $p$ ,  $q$  и  $r$  через пробел.  $p$  — номер преступления, которое было раскрыто, но Леше необходимо удалить.  $q$  и  $r$  — сложности двух вымышленных преступлений.

### Пример

стандартный ввод	стандартный вывод
11 -5 -4 -3 -2 -1 0 1 2 3 4 5	Possible 2 1 -5



## Задача I. Как может в казино колода разложена быть в другом порядке?

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

Недавно вы устроились дилером в казино. В вашу задачу входит разложение карт в случайном порядке. В казино, в котором вы работаете, используется следующий алгоритм перемешивания новой колоды карт:

```
integer seed;

integer rnd() {
    seed = (seed * 1103515245 + 12345) mod pow(2, 32);
    return seed;
}

array shuffle(array a, integer s) {
    seed = s;
    for i from 0 to a.size-1 do swap(a[i], a[rnd() mod (i + 1)]);
    return a;
}
```

В качестве параметра  $a$  передается упорядоченная колода из  $n$  карт (карты пронумерованы от 0 до  $n - 1$ ). В качестве параметра  $s$  передается случайное 32-битное целое беззнаковое число.

Только кажется этот алгоритм был скомпрометирован. Вы уже перемешали колоду и готовы к новой раздаче, когда узнали это. Случайный параметр перемешивания, к сожалению, никто никуда не записал в целях безопасности. Руководство предложило выход: перемешать колоду еще раз! Так, используя `seed`, который остался в памяти компьютера после первого перемешивания, перемешайте данную вам перемешанную колоду еще раз.

Каково будет удивление игрока, когда колода, заряженная в киоске будет разложена в другом порядке!

### Формат входных данных

В первой строке входного файла содержится одно целое число  $n$  — размер колоды.

Во второй строке содержатся  $n$  разделенных пробелом целых чисел  $a_0, a_1, \dots, a_{n-1}$  — перемешанная колода карт. Все числа  $a_0, a_1, \dots, a_{n-1}$  — различны.

$$1024 \leq n \leq 1\,048\,576$$

$$0 \leq a_0, a_1, \dots, a_{n-1} \leq n - 1$$

### Формат выходных данных

Выведите одну строку содержащую  $n$  целых чисел через пробел — колоду, перемешанную повторно.

### Пример

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

### Замечание

Пример в условии приведен только для ознакомления с форматом ввода и вывода и не будет использоваться в наборе тестов при тестировании посылок.

## Задача J. Доставка в городе будущего

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

Город будущего — идеальный прямоугольник разбитый на  $n \times m$  единичных квадратов, в каждом из которых расположено здание. Для упрощения схемы города, названия зданий на ней сокращены до одной первой буквы.

Транспорт будущего достаточно быстрый и позволяет произвести мгновенный перелет от здания к зданию, если удовлетворены следующие особенности:

- Эти здания находятся в одной строке или столбце схемы.
- Названия этих зданий совпадают по первой букве.
- На пути между этими зданиями есть хотя бы еще одно здание, которое также совпадает по первой букве названия.

Например: **abbbba** — невозможно, так как не выполняется 3-й пункт, **ababa** — возможно, так как есть еще одно здание **a**.

Доставка будущего — совершенно новый сервис, который пользуется транспортом будущего в городе будущего и доставляет пищу будущего. Вам поручено произвести последнюю доработку в интеллектуальной системе доставки будущего — проверку зоны доставки клиента. Так, вам будет дано  $q$  запросов  $\langle x_1, y_1, x_2, y_2 \rangle$ , а вы, пожалуйста, проверьте, возможно ли мгновенно доставить пищу транспортом будущего от здания  $\langle x_1, y_1 \rangle$  к зданию  $\langle x_2, y_2 \rangle$ . Мгновенная доставка может осуществляться с помощью нескольких мгновенных переходов.

### Формат входных данных

В первой строке входного файла содержатся два целых числа  $n, m$  — размеры схемы.

Каждая из последующих  $n$  строк содержит  $m$  маленьких букв латинского алфавита — первые буквы названий зданий.

Следующая строка содержит одно целое число  $q$  — количество запросов.

Каждая из последующих  $q$  строка содержит 4 разделенных пробелом целых числа  $x_1, y_1, x_2, y_2$  — параметры запроса.

$$1 \leq n, m \leq 1000$$

$$1 \leq q \leq 10^6$$

$$1 \leq x_1, x_2 \leq n$$

$$1 \leq y_1, y_2 \leq m$$

### Формат выходных данных

Выведите  $q$  строк,  $i$ -я из которых «Yes» или «No» — возможно ли доставить пищу в  $i$ -м запросе.

### Пример

стандартный ввод	стандартный вывод
3 3	No
aaa	No
aaa	No
aaa	Yes
5	Yes
1 1 1 2	
2 2 1 1	
2 2 1 2	
3 3 1 1	
3 2 1 2	

## **Замечание**

В первом примере все здания имеют одну и ту же первую букву в названии, однако доставить пиццу транспортом будущего не всегда возможно.

## Задача К. Абсолютная абсолютность массива

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

Вам дан массив  $a_i$  длины  $n$ , состоящий только из нулей и единиц. Назовем абсолютностью такого массива некоторое целое число  $s = |c_0 - c_1|$ , где  $c_0$  и  $c_1$  — количество нулей и единиц в массиве соответственно. Абсолютной абсолютностью массива будем называть некоторое целое число  $s_A$  равное максимальной абсолютности среди всех массивов, полученных путем замены в исходном массиве значений на непрерывном подотрезке (в том числе нулевой длины) на противоположные: единицы — на ноль, нуля — на единицу.

У вас есть  $q$  операций изменения значения определенного элемента в массиве на противоположное. После каждой операции требуется определить абсолютную абсолютность массива.

### Формат входных данных

В первой строке задано два целых числа  $n$  и  $q$  — длина массива и количество операций соответственно.

Во второй строке задано  $n$  целых чисел  $a_i$  — значения соответствующих элементов в массиве. Могут быть только нули и единицы.

В следующих  $q$  строках задано одно целое число  $p_i$  — номер элемента в массиве, значение которого нужно заменить на противоположное.

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

### Формат выходных данных

Выведите  $q$  строк. В  $i$ -й строке выведите одно целое число  $s_A$  — абсолютную абсолютность массива, после применения первых  $i$  операций.

### Пример

стандартный ввод	стандартный вывод
6 4	6
1 0 1 0 0 1	6
2	4
6	4
5	
1	

### Замечание

После первой операции, чтобы получить максимальную абсолютность массива, необходимо изменить значения  $a_4$  и  $a_5$  на 1.

## Задача L. Разделение кроликов

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

У Валеры есть  $n$  кроликов, пронумерованных от 1 до  $n$ . Так как их численность быстро растет, он хочет разделить их на две группы: группа номер 0 и группа номер 1.

Обозначим  $f(x) = y$  — кролик с номером  $x$  относится к группе номер  $y$ . Например,  $f(3) = 1$  означает, что кролик с номером 3 относится к группе номер 1. Тогда, если есть два кролика с номерами  $a$  и  $b$ , такими, что  $a$  делится на  $b$ , то должно выполняться следующее условие:  $f(a) = f(b) \text{ OR } f(\frac{a}{b})$ , где  $OR$  — операция побитового "ИЛИ". Невыполнение этого условия приведет к войне между двумя группами.

Для  $m$  кроликов Валера уже определил группу, к которой будет относиться каждый из них, однако дальше он просит вашей помощи. Помогите ему найти количество способов разделить кроликов на две группы так, чтобы они не начали войну, а кролики, для которых Валера определил группу, были в соответствующих группах. Так как это число может быть достаточно большим, Валеру интересует остаток от деления этого числа на  $10^9 + 7$ .

### Формат входных данных

В первой строке содержатся два целых числа, разделенных пробелами  $n$  и  $m$  — общее количество кроликов и количество кроликов, для которых Валера уже определил группу, соответственно.

В следующих  $m$  строках содержатся по два целых числа  $x_i$  и  $y_i$ . Каждая из этих строк означает, что кролик с номером  $x_i$  должен находиться в группе номер  $y_i$ . Гарантируется, что все  $x_i$  различны.

$$1 \leq n \leq 10^6$$

$$1 \leq m \leq 18$$

$$1 \leq x_i \leq n$$

$$0 \leq y_i \leq 1$$

### Формат выходных данных

В единственной строке выведите одно число — количество способов разбить кроликов на две группы, которые удовлетворяют заданным условиям, по модулю  $10^9 + 7$ .

### Примеры

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