

## Задача А. Тариф

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

Девочка Катя подключилась к тарифу «Очень выгодный», на котором можно только звонить.

Все входящие звонки бесплатны. В случае исходящего звонка  $k_1$  первые секунд звонка стоят  $p_1$  копеек. За следующие  $k_2$  секунд Катя платит по  $p_2$  копеек за секунду, а все остальное время девочка платит по  $p_3$  копеек за секунду. Деньги снимаются мгновенно после каждой секунды. Как только баланс становится неположительным, связь обрывается. Известно, что Катя положила  $N$  копеек на счет, чтобы поговорить со своим лучшим другом.

Посчитайте, сколько максимально секунд Катя сможет наслаждаться беседой.

### Формат входных данных

Во входном файле записаны через пробел 6 целых чисел:  
 $0 \leq N \leq 1000000$ ,  $1 \leq k_1, k_2 \leq 1000000$ ,  $1 \leq p_1, p_2, p_3 \leq 1000000$ .

### Формат выходных данных

Выведите одно число — максимальную продолжительность звонка в секундах.

## Задача В. Выставка

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

Известный художник Блобс устраивает выставку своих работ в местной картинной галерее. Он хочет представить на выставке  $k$  своих работ, но в галерее есть всего лишь  $n$  держателей для картин. Во время подготовки выяснилось, что держатели рассчитаны на картины разного веса. На держатель с номером  $i$  может быть установлена картина весом не более  $d_i$  граммов. По этим причинам может случиться так, что не удастся выставить все картины. В связи с этим обстоятельством Блобс определил для каждой картины два значения: красоту —  $a_i$  и вес в граммах —  $w_i$ . Помогите Блобсу выбрать множество картин для выставки, так чтобы максимизировать суммарную красоту всех выставленных картин.

Напишите программу, которая найдет требуемое множество картин. Если оптимальных ответов несколько, разрешается вывести любой.

### Формат входных данных

В первой строке входного файла содержатся два числа разделенных пробелом:  $n$  и  $k$ .

Вторая строка содержит  $n$  разделенных пробелом чисел —  $d_1, d_2, \dots, d_n$ . В следующих  $k$  строках по два числа —  $a_j$  и  $w_j$ .

Ограничения:

$$1 \leq n \leq k \leq 10\,000$$

$$1 \leq a_j, w_j, d_i \leq 1\,000\,000, 1 \leq i, j \leq n$$

### Формат выходных данных

Выведите  $n$  чисел,  $i$ -е из которых обозначает номер картины, выставленной на  $i$ -ом держателе (первая картина имеет номер 1). Если держатель пуст, выведите 0 на соответствующей позиции. Картины должны быть занумерованы в том порядке, в котором они встречаются во входном файле.

## Задача С. Родственные связи

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

Еще недавно королевство было совсем маленьким, все всех знали, и определение родственных связей не представляло никакой проблемы. Так сложилось, что в последние годы население резко выросло, и стало сложно определить кто кому кем приходится.

Мало кто знает, но у каждого зверя есть паспорт, в котором указан его номер — целое неотрицательное число. Паспортная система королевства хороша, но не идеальна, поэтому эти номера совсем не обязательно уникальны. Хороша она, собственно, тем, что если в десятичной записи номеров паспортов двух зверей есть хотя бы одна общая цифра, то они являются родственниками. Например, звери с номерами 47 и 107 — родственники, а с номерами 74 и 931 — нет.

Король Лев, как настоящий правитель, хочет знать все о своих подданных, поэтому просит вас посчитать количество различных пар зверей, которые являются родственниками.

### Формат входных данных

Первая строка входного файла содержит число  $N$  ( $1 \leq N \leq 500000$ ) — количество зверей в королевстве.

Вторая строка содержит  $N$  целых неотрицательных чисел, не превышающих  $10^9$ , разделенных пробелами — номера зверей.

### Формат выходных данных

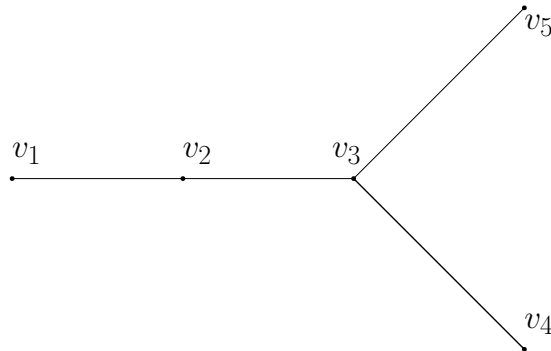
Выведите единственное число — ответ на задачу.

## Задача D. Буквы Y в дереве

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

Вам дано дерево (связный неориентированный граф без циклов). Посчитайте количество «букв Y» в нём — таких наборов  $\{v_1, v_2, v_3, v_4, v_5\}$  из пяти разных вершин дерева, что следующие пары вершин соединены ребром:  $v_1$  и  $v_2$ ,  $v_2$  и  $v_3$ ,  $v_3$  и  $v_4$ ,  $v_3$  и  $v_5$ . При этом наборы, отличающиеся перестановкой вершин  $v_4$  и  $v_5$ , считаются одинаковыми.

Другими словами, вас просят посчитать, сколько раз в дереве встречается следующая картинка:



### Формат входных данных

В первой строке задано число  $n$  ( $2 \leq n \leq 10^5$ ) — количество вершин дерева. Во второй строке через пробел заданы числа  $p_2, p_3, \dots, p_n$  ( $1 \leq p_i \leq i-1$  для каждого  $i$  от 2 до  $n$  включительно). Эти числа означают, что для каждого  $i$  от 2 до  $n$  включительно в дереве есть ребро между вершиной  $i$  и вершиной  $p_i$ , а никаких других рёбер в дереве нет. Можно доказать, что любой граф, который можно задать таким образом, является деревом.

### Формат выходных данных

Выведите единственное число — количество «букв Y» в дереве.

## Задача E. ПДД

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

В столице одной небольшой страны очень сложная ситуация. Многокилометровые пробки буквально парализовали движение в городе, и власти на многих улицах ввели одностороннее движение, не анализируя, можно ли будет теперь проехать из любого места в городе в любое другое, не нарушая правила. Транспортная система столицы представляет собой  $N$  площадей, соединенных  $M$  полосами для движения, в том числе круговыми полосами, проходящими по площади. Каждая полоса предназначена для движения только в одну определенную сторону. При этом на магистралях есть полосы, направленные как в одну, так и в другую сторону. По круговой полосе можно двигаться только внутри площади и только против часовой стрелки.

Власти города на каждой полосе разместили видеокамеру, поэтому если Иннокентий едет по встречной полосе (при ее наличии) или, в случае одностороннего движения, в сторону противоположную предписанной знаками, то после поездки против правил по каждой из полос ему придется заплатить штраф в размере одной тысячи тугриков этой страны.

Иннокентий, который торопится купить кафельную плитку со скидкой, решил доехать до магазина в любом случае, даже если для этого придется нарушать правила. Но он хочет выбрать такой маршрут движения, суммарный штраф на котором минимален.

Иннокентий еще не решил, откуда именно и в какой магазин он собирается ехать, поэтому ему необходимо ответить на несколько вопросов вида «Какой минимальный штраф надо заплатить, чтобы добраться из пункта  $A$  в пункт  $B$ ?». Отвечая на потребности жителей столицы, известная поисковая система Индекс разрабатывает соответствующий сервис.

Так как многие из вас рано или поздно будут проходить собеседование на работу в эту фирму, продемонстрируйте, что вы тоже умеете решать эту задачу.

### Формат входных данных

В первой строке входных данных содержатся два числа  $N$  и  $M$  — количество площадей и полос движения в городе соответственно ( $1 \leq N \leq 5000, 1 \leq M \leq 10\,000$ ). Далее содержатся описания полос, по которым движение разрешено. Каждая полоса описывается номерами двух площадей  $a_i, b_i$  ( $1 \leq a_i, b_i \leq N$ ), которые она соединяет. Движение разрешено в направлении от первой из указанных площадей ко второй.

В следующей строке содержится одно число  $K$  — количество вопросов у Иннокентия ( $1 \leq K \leq 10\,000, N \cdot K \leq 2 \cdot 10^7$ ). В следующих строках описываются вопросы, каждый вопрос описывается номерами двух площадей  $a_i, b_i$  ( $1 \leq a_i, b_i \leq N$ ), между которыми требуется найти самый дешевый путь. Путь необходимо проложить от первой из указанных площадей ко второй.

### Формат выходных данных

Для каждого вопроса выведите одно число — искомый минимальный размер штрафа в тысячах тугриков. В случае, если пути между выбранной парой площадей не существует, выведите «-1».

## Задача F. Ор до гор

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

Даша собирается пойти в поход в Хибины. Хибины представляют из себя горный массив, состоящий из  $n$  гор, пронумерованных от 1 до  $n$ . Гора с номером  $i$  имеет высоту  $h_i$  метров. Перед тем, как пойти в поход, Даше нужно составить маршрут. Маршрут похода задается двумя целыми числами  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ). В течение похода Даша взберётся на все горы, номера которых принадлежат отрезку чисел от  $l$  до  $r$  включительно. Более формально, в маршрут, заданный числами  $l$  и  $r$ , входят все горы с номерами  $m$ , для которых выполнено условие  $l \leq m \leq r$ . Даша увлекается программированием и очень любит целое число  $k$ , поэтому она хочет, чтобы побитовый  $OR$  высот гор из маршрута был равен  $k$ . Помогите Даше найти такой маршрут, или сообщите, что его не существует.

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n, k$  ( $1 \leq n \leq 10^6, 1 \leq k \leq 10^9$ ) — количество гор в Хибинах и любимое число Даше. Вторая строка входных данных содержит  $n$  натуральных чисел  $h_i$  — высоты гор ( $1 \leq h_i \leq 10^9$ ).

### Формат выходных данных

Если маршрута, удовлетворяющего условию, не существует, выведите «NO». Иначе, в первой строке выведите «YES», а во второй — числа  $l$  и  $r$ , задающие подходящий маршрут. Если правильных ответов несколько, выведите любой из них.

### Пример

стандартный ввод	стандартный вывод
5 7 1 2 3 4 5	YES 1 4

### Замечание

Напомним, что побитовый  $OR$  набора чисел — это число, каждый двоичный бит которого равен результату операции  $OR$ , примененной к соответствующим двоичным битам исходных чисел.

Результат операции  $OR$  для чисел, равных 0 или 1 определяется следующим образом:

- Если все числа равны 0, то результат операции тоже равен 0.
- В противном случае, результат операции равен 1.

## Задача G. Тест на терпение

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

Как вы знаете, Энакину не хватает терпения, поэтому мастер Оби-Ван решил дать много одинаковых задачек своему ученику. А именно, он дал ему  $n$  троек чисел  $a_i, b_i, c_i$  и попросил найти для каждой тройки два числа  $x_i$  и  $y_i$  таких, что побитовое И этих двух чисел равно  $a_i$ , побитовое ИЛИ этих чисел равно  $b_i$ , а их разность  $x_i - y_i$  равна  $c_i$ . У Энакина нет времени на решение задач, поэтому он просит вас сделать это задание вместо него. Учтите, что Оби-Ван торопился на совет джедаев, поэтому среди троек могут быть такие, для которых решение не существует.

### Формат входных данных

В первой строчке дано целое число  $n$  — количество троек ( $1 \leq n \leq 10000$ ).

В следующих  $n$  строчках даны описания троек — три целых числа  $a_i, b_i, c_i$  ( $0 \leq a, b, c \leq 10^9$ ).

### Формат выходных данных

Для каждой тройки определите, существует ли два числа, удовлетворяющих условиям.

Если существует, то выведите в одной строчке два целых числа  $x_i, y_i$  — ответ для тройки  $i$ . Если существует несколько правильных пар чисел, являющихся ответом, разрешается вывести любую.

Если не существует, то выведите в одной строчке  $-1$ .

### Пример

стандартный ввод	стандартный вывод
3	12 10
8 14 2	6 3
2 7 3	-1
8 13 4	

### Замечание

Побитовое И двух неотрицательных целых чисел  $a$  и  $b$  определяется так. Сначала два числа записываются в двоичной системе счисления одно над другим так, чтобы последняя цифра второго числа оказалась точно под последней цифрой первого числа. Далее, если записи чисел оказались неравной длины, то более короткое из двух чисел дополняется слева нулями, пока цифр в числах не станет поровну. Далее под этими двумя числами записывается третье двоичное число по следующим правилам: если на  $i$ -м месте в первом и втором числе стоят 1, то в третьем числе на  $i$ -м месте записывается 1; в противном случае там записывается 0. Построенное таким образом третье число и является двоичной записью побитового И данных двух чисел.

Например, рассмотрим числа 17 и 71. В двоичном виде они записываются как 10001 и 1000111. Дополним первое число двумя нулями слева, чтобы в нём стало семь цифр, как и во втором числе: 0010001. Напишем под парой цифр 1, если они обе равны 1, и 0 иначе:

```
0010001
1000111
0000001
```

Если стереть лидирующие нули, получится 1 — двоичная запись числа 1. Таким образом,  $17 \text{ AND } 71$  — побитовое И чисел 17 и 71 — равно 1.

Похожим образом определяется побитовое ИЛИ. Мы так же выписываем числа, дополняем нулями, но под двумя цифрами ставим 1 в случае, если хотя бы одна из цифр равна 1, а иначе ставим 0. В частности, для тех же чисел 17 и 71 имеем:

```
0010001
1000111
1010111
```

Полученная двоичная запись представляет число  $2^6 + 2^4 + 2^2 + 2^1 + 2^0 = 87$ . Таким образом,  $17 \text{ OR } 71$  — побитовое ИЛИ чисел 17 и 71 — равно 87.

В языке Pascal побитовые И и ИЛИ вычисляются с помощью команд  $a \text{ and } b$  и  $a \text{ or } b$ , а в C++, Java, Python — с помощью  $a \& b$  и  $a | b$ .

## Задача Н. Изменения массива

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

На перемене Ваня зашел в класс и увидел на доске массив из  $n$  целых неотрицательных  $k$ -битных чисел  $a_1, a_2, \dots, a_n$ . Число  $x$  называется  $k$ -битным, если  $0 \leq x \leq 2^k - 1$ .

Естественно, Ваня не удержался и стал менять числа, написанные на доске. Для того, чтобы никто ничего не заметил, Ваня разрешил себе сделать несколько следующих изменений: выбрать индекс массива  $i$  ( $1 \leq i \leq n$ ) и заменить число  $a_i$  на число  $\bar{a}_i$ . Будем обозначать за  $\bar{x}$  для  $k$ -битного числа  $x$  такое  $k$ -битное число, что все его  $k$  битов отличаются от соответствующих битов числа  $x$ .

Ване очень не нравится число 0. Поэтому ему нравятся такие отрезки  $[l, r]$  ( $1 \leq l \leq r \leq n$ ), что  $a_l \oplus a_{l+1} \oplus \dots \oplus a_r \neq 0$ , где  $\oplus$  обозначает операцию побитового исключающего ИЛИ. Определите, какое максимальное количество таких отрезков он сможет получить, выполнив несколько (возможно, ноль) операций, описанных выше.

### Формат входных данных

В первой строке входных данных находятся два целых числа  $n$  и  $k$  ( $1 \leq n \leq 200\,000$ ,  $1 \leq k \leq 30$ ).

В следующей строке находятся  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 2^k - 1$ ), разделённых пробелами — массив из  $k$ -битных чисел.

### Формат выходных данных

Выведите одно целое число — максимальное возможное количество отрезков с побитовым исключающим ИЛИ, не равным 0, которое можно получить, сделав несколько (возможно, 0) операций, описанных в условии.

### Пример

стандартный ввод	стандартный вывод
6 3 1 4 4 7 3 4	19

### Замечание

В первом тесте можно не делать операций, тогда мы получим, что у нас 5 отрезков, которые нравятся Ване. Если сделать операцию с  $i = 2$ , то получится массив  $[1, 0, 0]$ , так как  $\bar{3} = 0$  при  $k = 2$ . У такого массива будет 3 подотрезка, которые нравятся Ване. Можно получить массив с 5 подотрезками, которые нравятся Ване, если сделать операцию с  $i = 3$  и операцию с  $i = 2$ . Тогда получится массив  $[1, 0, 3]$ . Можно доказать, что 6 и более отрезков, которые будут нравиться Ване, получить невозможно.

Во втором тесте, чтобы получить 19 подотрезков, которые нравятся Ване, можно сделать 4 операции с  $i = 3$ ,  $i = 4$ ,  $i = 5$  и  $i = 6$ . Получится массив  $[1, 4, 3, 0, 4, 3]$ .



## Задача I. Игра с модулем

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

### Это интерактивная задача.

Вася и Петя собираются сыграть в следующую игру: Петя загадает некоторое целое число  $a$ . После этого Вася должен будет отгадать его, задав несколько вопросов. Он может сказать пару целых неотрицательных чисел  $(x, y)$ . В ответ Петя скажет:

- “x”, если  $(x \bmod a) \geq (y \bmod a)$ .
- “y”, если  $(x \bmod a) < (y \bmod a)$ .

Операция  $(x \bmod a)$  обозначает операцию взятия остатка  $x$  при делении на  $a$ .

Вася должен отгадать число  $a$  за **не более, чем 60 вопросов**.

**Гарантируется, что** Петя загадал число, удовлетворяющее неравенству  $1 \leq a \leq 10^9$ .

Помогите Васе играть в эту игру и напишите программу, которая будет угадывать число  $a$ .

## Протокол взаимодействия

Вам нужно будет сыграть несколько игр.

Перед началом каждой игры вы должны считать строку:

- “start” (без кавычек) — начало новой игры.
- “mistake” (без кавычек) — в предыдущей игре был назван неверный ответ. Считав такую строку ваша программа должна завершиться и тогда она получит вердикт “Неверный ответ”.
- “end” (без кавычек) — все игры закончены. Считав такую строку, ваша программа должна завершиться.

После считывания строки “start” (без кавычек) начинается очередная игра.

Сначала ваша программа может сделать несколько запросов вида спросить пару целых чисел  $(x, y)$ . Можно спрашивать только числа, удовлетворяющие неравенствам  $0 \leq x, y \leq 2 \cdot 10^9$ . Для этого нужно вывести “? x y” (без кавычек). В ответ ваша программа получит один из символов:

- “x” (без кавычек), если  $(x \bmod a) \geq (y \bmod a)$ .
- “y” (без кавычек), если  $(x \bmod a) < (y \bmod a)$ .
- “e” (без кавычек) — вы задали больше 60 вопросов. Считав такой символ, ваша программа должна завершиться и тогда она получит вердикт “Неверный ответ”.

После того, как вы зададите несколько вопросов ваша программа должна вывести ответ в формате “! a” (без кавычек). Выведенное вами число  $a$  должно удовлетворять неравенству  $1 \leq a \leq 10^9$ . Гарантируется, что загаданное число  $a$  удовлетворяло этому неравенству. На этом текущая игра закончится.

Напомним, что ваша программа не может задать больше, чем 60 вопросов во время одной игры.

Если ваша программа не завершится после считывания “mistake” (без кавычек), “end” (без кавычек) или “e” (без кавычек), то она может получить любой вердикт, потому что будет считывать данные из закрытого потока ввода. Также, если ваша программа задаст вопрос или выведет ответ в неверном формате, она может получить любой вердикт. **Будьте внимательны.**

**Не забудьте сбрасывать буфер вывода после того, как зададите вопрос или выведете ответ.**

Чтобы сбросить буфер вывода вы можете использовать:

- `fflush(stdout)` в C++.

- `System.out.flush()` в Java.
- `stdout.flush()` в Python.
- `flush(output)` в Pascal.
- Прибегните к документации других языков.

Гарантируется, что будет сыграно не меньше 1 и не больше, чем 100 игр.

## Пример

стандартный ввод	стандартный вывод
start	? 0 0
x	? 10 1
x	! 1
start	? 0 0
x	? 3 4
x	? 2 5
y	! 2
start	? 2 4
x	? 2 5
x	? 3 10
y	? 9 1
y	! 3
end	

## Замечание

В тесте из условия будет сыграно 3 игры, в которых будут загаданы числа 1, 2 и 3.

В первой игре ответом на любой вопрос будет “x” (без кавычек), так как  $(x \bmod 1) = 0$  для всех целых  $x$ .

Во второй игре, если спросить пару  $(0, 0)$ , то ответом будет “x” (без кавычек), так как  $(0 \bmod 2) \geq (0 \bmod 2)$ . А если, например, спросить пару  $(2, 5)$ , то ответом будет “y” (без кавычек), так как  $(2 \bmod 2) < (5 \bmod 2)$ , потому что  $(2 \bmod 2) = 0$  и  $(5 \bmod 2) = 1$ .