

# Смотреть условия до команды волонтеров запрещено!

Во вторник, 29 мая, на этих же задачах будет проведён раунд на платформе Codeforces.

Пожалуйста, не обсуждайте задачи ни с кем, кроме участников соревнования, до 30 мая.

Разбор задач в письменном виде можно будет найти на платформе Codeforces 30 мая.

Ссылка на вход в тестирующую систему:  
<https://official.contest.yandex.ru/contest/8290/>

## Задача А. Проблемы андроидов

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

На дворе 2118 год. Давно в массовом производстве андроиды, которые делают за людей всю работу. Однако чтобы андроиды могли решать нестандартные задачи, им нужно учиться в школе. Как людям раньше.

Все школьные проблемы никуда не исчезли, в школах все также задирают учеников, которые не похожи на других. Вася-8800 — бюджетный андроид, производит его малоизвестная компания, и дизайн его не выверен до мелочей. Технические характеристики Васи тоже не совершенны. Другие андроиды выбрали его целью своих не всегда добрых шуток.

Любимая шутка недругов Васи — заставить его сравнивать  $x^y$  и  $y^x$ . Другие андроиды справляются с этой задачей за миллисекунды, а Васе не хватает памяти, чтобы сохранить такие большие числа.

Помогите Васе! Напишите для Васи быструю программу, которая сравнивает  $x^y$  и  $y^x$ , и тогда, возможно, другие андроиды станут его уважать.

### Формат входных данных

В единственной строке записаны два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq 10^9$ ).

### Формат выходных данных

Если  $x^y < y^x$ , выведите “<” (без кавычек). Если  $x^y > y^x$ , выведите “>” (без кавычек). Если же  $x^y = y^x$ , выведите “=” (без кавычек).

### Примеры

стандартный ввод	стандартный вывод
5 8	>
10 3	<
6 6	=
14 1	>
2 4	=
987654321 123456987	<

## Задача В. Курсы по выбору

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

В Высшей Школе Экономики каждый студент 3-го и 4-го курса должен выбрать себе два курса по выбору, по одному на каждый семестр. Процесс записи на курсы происходит онлайн. Опишем процесс распределения студентов по курсам одного семестра подробнее.

Деканат оглашает список вариантов, в нём  $m$  курсов. Каждый курс имеет название, минимальное количество студентов для того, чтобы курс состоялся, и максимальное количество студентов, которые могут записаться на этот курс.

После этого каждый студент заполняет онлайн-форму записи. В этой форме студент ранжирует все предложенные курсы по степени своего желания посещать их: от наиболее желаемых к наименее желаемым.

У каждого студента есть рейтинг. Для простоты будем считать, что все рейтинги являются целыми неотрицательными числами и попарно различны.

На первом этапе определяются курсы, которые состоятся. Для этого предполагается, что каждый студент будет посещать наиболее желаемый для себя курс. Затем для каждого курса подсчитывается, сколько студентов будут посещать этот курс, и, если количество студентов не менее минимального количества студентов для данного курса, то курс считается состоявшимся, иначе — не состоявшимся. Все студенты, которые первым приоритетом поставили курс, который не состоялся, сразу отправляются на второй этап. Те же студенты, которые первым приоритетом поставили курс, который состоялся, записываются на этот курс. Однако если количество студентов, выбравших некий курс, превышает максимальное количество студентов для данного курса, то приоритет получают те студенты, рейтинг которых выше. Студенты с низким рейтингом, которые не поместились на курс своего первого приоритета, также отправляются на второй этап.

На втором этапе рассматриваются студенты, для которых ещё не определён курс, в порядке убывания их рейтинга. Для каждого студента выбирается курс, который:

- состоялся;
- ещё имеет свободные места;
- среди всех курсов, удовлетворяющих первым двум условиям, выбирается курс, который наиболее желаем для данного студента (находится выше всего в списке приоритетов).

Студент записывается на выбранный курс. Если таких курсов нет, то студента отчисляют (сурово, а что делать).

Ваша задача — реализовать эту систему.

### Формат входных данных

В первой строке записаны два целых положительных числа  $n$  и  $m$  — количество студентов и количество предлагаемых курсов ( $1 \leq n \leq 300$ ,  $1 \leq m \leq 20$ ).

В следующих  $m$  строках по одному в строке описываются курсы. Описание каждого курса имеет вид  $name_i \ minStudent_i \ maxStudent_i$  — название курса, минимальное количество студентов, необходимое, чтобы курс состоялся, максимальное количество студентов, которые могут посещать данный курс ( $1 \leq minStudent_i \leq maxStudent_i \leq n$ ). Название курса является непустой строкой, состоящей только из заглавных и строчных латинских букв и цифр, а его длина не превышает 20 символов.

В следующих  $n$  строках по одному в строке описываются студенты. Описание каждого студента имеет вид  $surname_i \ rating_i \ preferences_i$  — фамилия, рейтинг, список предпочтений в порядке от более желаемых к менее желаемым. Фамилия является непустой строкой, которая состоит только из заглавных и строчных латинских букв, а её длина не превышает 20.  $0 \leq rating_i \leq 10^5$ . В списке предпочтений встречаются только названия курсов и название каждого курса встречается там ровно один раз.

Гарантируется, что названия всех курсов различны. Гарантируется, что фамилии всех студентов различны. Гарантируется, что рейтинги всех студентов различны. Гарантируется, что не существует курса с названием “SynergyUniversity”.

### Формат выходных данных

В первой строке выведите названия всех состоявшихся курсов в алфавитном порядке.

Затем в  $n$  строках выведите для каждого студента его фамилию и название курса, на который его записали. Если студента отчислили, вместо названия курса выведите “SynergyUniversity” (без кавычек). Студентов выводите в алфавитном порядке их фамилий.

### Примеры

стандартный ввод	стандартный вывод
6 4 Python3 1 2 CPlusPlus 1 1 MachineLearning 1 2 Calculus 3 4 Grinberg 666 Calculus MachineLearning CPlusPlus Python3 Storozhenko 721 CPlusPlus Python3 Calculus MachineLearning Kutenin 540 CPlusPlus Calculus MachineLearning Python3 Musatkina 491 Python3 MachineLearning CPlusPlus Calculus Danilyuk 42 Calculus Python3 CPlusPlus MachineLearning Ryabinin 777 MachineLearning Python3 CPlusPlus Calculus	CPlusPlus MachineLearning Python3 Danilyuk SynergyUniversity Grinberg MachineLearning Kutenin Python3 Musatkina Python3 Ryabinin MachineLearning Storozhenko CPlusPlus
2 2 RealStudying 1 1 Memology 2 2 Lupa 0 Memology RealStudying Pupa 99999 RealStudying Memology	RealStudying Lupa RealStudying Pupa SynergyUniversity

## Задача С. Ярмарка

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

В Бйитландии собираются провести ярмарку. В Бйитландии  $n$  городов, которые соединены  $m$  двухсторонними дорогами, причём из любого города можно доехать до любого другого, передвигаясь только по дорогам.

В Бйитландии производят  $k$  различных товаров, причём каждый город специализируется на одном товаре. Чтобы ярмарка состоялась, на неё нужно привезти хотя бы  $s$  различных товаров. Чтобы привезти товары из города  $u$  в город  $v$  нужно потратить  $d(u, v)$  монет, где  $d(u, v)$  — длина кратчайшего пути между городами  $u$  и  $v$ . Длина пути — это количество дорог, которые входят в этот путь.

Организаторы ярмарки оплатят перевозку товаров, однако они сами могут выбрать, производителей из каких городов пригласить на ярмарку. Теперь организаторы хотят для каждого города посчитать, какое минимальное количество монет нужно потратить на перевозку товаров, чтобы провести ярмарку в этом городе.

### Формат входных данных

В первой строке записаны 4 целых числа  $n, m, k, s$  ( $1 \leq n \leq 10^5, 0 \leq m \leq 10^5, 1 \leq s \leq k \leq \min(n, 100)$ ) — количество городов, количество дорог, количество различных товаров, количество различных товаров необходимых для проведения ярмарки.

В следующей строке записаны  $n$  чисел  $a_i$  — номер товара, который производится в  $i$ -м городе ( $1 \leq a_i \leq k$ ). Гарантируется, что среди чисел  $a_i$  встречаются все числа от 1 до  $k$ .

В следующих  $m$  строках описываются дороги. Каждая дорога описывается парой  $u, v$  городов, которые она соединяет ( $1 \leq u, v \leq n, u \neq v$ ). Гарантируется, что между любой парой городов не более одной дороги. Гарантируется, что можно доехать из любого города в любой, двигаясь только по дорогам.

### Формат выходных данных

Выведите  $n$  чисел,  $i$ -е из которых — минимальное количество монет на перевозку товаров при проведении ярмарки в городе  $i$ . Разделяйте числа пробелами.

### Примеры

стандартный ввод	стандартный вывод
5 5 4 3 1 2 4 3 2 1 2 2 3 3 4 4 1 4 5	2 2 2 2 3
7 6 3 2 1 2 3 3 2 2 1 1 2 2 3 3 4 2 5 5 6 6 7	1 1 1 2 2 1 1

## Задача D. Петя и перестановки

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

Петя любит придумывать задачи со случайными данными. На этот раз ему понадобилась случайная перестановка. Он решил сгенерировать её следующим образом: он берет тождественную перестановку размера  $n$ , а потом  $3n$  раз берёт случайную пару различных элементов и меняет их местами. Лёша очень завидует Пете, и пытается ему во всем подражать. Лёша тоже придумал задачу со случайной перестановкой, и генерирует её как Петя, но меняет местами случайную пару элементов не  $3n$  раз, а  $7n + 1$ . Потому что это более случайно, понятно?!

Вам попал в руки тест, и вы хотите узнать, от какой он задачи.

### Формат входных данных

В первой строке вводится одно целое число  $n$  ( $10^3 \leq n \leq 10^6$ ).

Во второй строке вводится  $n$  различных целых чисел от 1 до  $n$  — перестановка размера  $n$  из теста.

Гарантируется, что все тесты, кроме примера из условия, сгенерированы следующим образом: Сначала выбирается  $n$  — размер перестановки. Затем случайно равномерно выбирается способ генерации — Петин или Лёшин. Затем выбранным способом генерируется сама перестановка.

### Формат выходных данных

Если тест сгенерирован способом Пети, выведите “Petr” (без кавычек). Если тест сгенерирован способом Лёши, выведите “Um\_nik” (без кавычек).

### Пример

стандартный ввод	стандартный вывод
5 2 4 5 1 3	Petr

### Замечание

Обратите внимание, что пример из условия не удовлетворяет ограничениям на  $n$  и приведён только для демонстрации формата ввода и вывода. Тем не менее, ваше решение должно проходить этот тест.

## Задача E. AND Graph

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Дано множество размера  $m$ , состоящее из целых неотрицательных чисел, каждое из которых строго меньше  $2^n$ . Построим на этих числах неориентированный граф следующим образом: соединим  $x$  и  $y$ , если и только если  $x \& y = 0$ . Здесь  $\&$  — операция побитового И. Посчитайте количество компонент связности в таком графе.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $0 \leq n \leq 22$ ,  $1 \leq m \leq 2^n$ ).

Во второй строке через пробел перечислены  $m$  чисел  $a_i$  — элементы множества ( $0 \leq a_i < 2^n$ ). Все  $a_i$  различны.

### Формат выходных данных

Выведите одно целое число — количество компонент связности.

### Примеры

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

## Задача F. Идеальное кодирование

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

Вы работаете аналитиком в компании, которая разрабатывает новую систему хранения больших данных. В системе нужно будет хранить  $n$  различных объектов. Разумеется, каждому объекту нужно присвоить уникальный ID.

Сначала выбираются параметры системы — целые числа  $m \geq 1$  и  $b_1, b_2, \dots, b_m$ . ID каждого объекта — это массив целых чисел  $[a_1, a_2, \dots, a_m]$ , причём выполнено  $1 \leq a_i \leq b_i$ .

Разработчики говорят, что затраты на разработку будут пропорциональны  $\sum_{i=1}^m b_i$ . Вас попросили подобрать параметры  $m$  и  $b_i$  так, чтобы система могла присвоить  $n$  объектам различные ID, а затраты на разработку были минимальны. Обратите внимание, что необязательно использовать все доступные ID.

### Формат входных данных

В единственной строке записано одно целое положительное число  $n$ . Длина десятичной записи  $n$  не превосходит  $1.5 \cdot 10^6$ .

### Формат выходных данных

Выведите одно целое число — минимальное значение  $\sum_{i=1}^m b_i$ .

### Примеры

стандартный ввод	стандартный вывод
36	10
37	11
12345678901234567890123456789	177

## Задача G. Задача Принца

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Действующими лицами этой задачи будут герои какого-нибудь недавно вышедшего в прокат фильма. Кажется, в последнее время много мемов по «Мстители: Война бесконечности». Я не смотрел ни одной части, поэтому лишь смутно знаю героев. Впрочем, это не мешает мне использовать их в задаче. Пусть героями нашей задачи будут Танос и Доктор Стрендж. Итак, Танос и Доктор Стрендж занимаются своими супергеройскими и суперзлодейскими делами, как вдруг они наталкиваются на обычную задачу по спортивному программированию.

Есть дерево размера  $n$ .

В каждой вершине  $v$  записано целое положительное число  $a_v$ .

Нужно ответить на  $q$  запросов.

Запрос задаётся в виде  $u v x$ .

Нужно посчитать  $\prod_w \gcd(x, a_w) \bmod (10^9 + 7)$ , где  $w$  в произведении пробегает все вершины пути из  $u$  в  $v$ .

Обратите внимание, что сами числа в вершинах не изменяются.

Думаю, вам интереснее было бы смотреть на супергеройские похождения Доктора Стренджа и Таноса, чем на то, как они решают задачку. Поэтому вам предлагается решить задачу вместо них.

### Формат входных данных

В первой строке записано одно целое число  $n$  — размер дерева ( $1 \leq n \leq 10^5$ ).

В следующих  $n-1$  строках описываются рёбра дерева — каждое ребро описывается двумя своими концами  $u_i v_i$  ( $1 \leq u_i, v_i \leq n$ ).

В следующей строке записаны  $n$  целых положительных чисел  $a_v$  ( $1 \leq a_v \leq 10^7$ ).

В следующей строке записано одно целое положительное число  $q$  — количество запросов ( $1 \leq q \leq 10^5$ ).

Наконец, в  $q$  строках описываются запросы в формате  $u_i v_i x_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq x_i \leq 10^7$ ).

### Формат выходных данных

Выведите  $q$  чисел в отдельных строках — ответы на запросы в том порядке, в котором они давались во входных данных.

### Пример

стандартный ввод	стандартный вывод
4	36
1 2	4
1 3	1
1 4	
6 4 9 5	
3	
2 3 6	
2 3 2	
3 4 7	

## Задача Н. Oppa Funcan Style Remastered

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Наверняка вы видели безумные клипы южнокорейского рэпера PSY, такие как «Gangnam Style», «Gentleman» и «Daddy». Вы также могли слышать, что PSY пару лет назад записывал клип «Oppa Funcan Style» (к сожалению, мы не смогли найти этот клип на просторах сети; мы не исключаем, что виноват Ромкоснадзор). Мы напомним вам, как выглядел этот хит:

На площадке есть  $n$  платформ, пронумерованных целыми числами от 1 до  $n$ , на  $i$ -й платформе стоит танцор с номером  $i$ . Далее, каждую секунду все танцоры, стоявшие на платформе с номером  $i$ , перепрыгивают на платформу с номером  $f(i)$ . Правило перемещения  $f$  выбирается заранее и не меняется на протяжении клипа.

Продолжительность клипа составляла  $k$  секунд, и правило  $f$  было выбрано таким образом, чтобы в конце клипа все танцоры оказались на своих изначальных местах. Это позволило зациклить клип и собрать ещё больше лайков.

PSY знает, что сейчас всё большую популярность набирают улучшенные версии старых произведений искусства, поэтому он решил выпустить remastered-версию своего клипа двухлетней давности.

В данном случае «улучшенная версия» подразумевает ещё больше безумия, и количество платформ может достигать  $10^{18}$ ! Однако режиссёр клипа говорит, что если какой-то танцор будет оставаться на одной и той же платформе, то зритель заскучает и сразу выключит клип. Поэтому для всех  $x$  от 1 до  $n$  должно быть выполнено  $f(x) \neq x$ .

Немалая часть успеха классического клипа была в том, что его удалось зациклить, поэтому для remastered-версии это также должно быть выполнено.

PSY ещё не определился с точным количеством платформ и продолжительностью клипа, поэтому он просит вас для разных вариантов определить, существует ли подходящее правило  $f$ .

### Формат входных данных

В первой строке записано одно число  $t$  — количество вариантов  $n$  и  $k$ , которые необходимо проверить ( $1 \leq t \leq 10^4$ ).

В следующих  $t$  строках записаны сами варианты: два числа  $n$  и  $k$  ( $1 \leq n \leq 10^{18}$ ,  $1 \leq k \leq 10^{15}$ ).

**Гарантируется, что количество различных  $k$  в одном тесте не превосходит 50.**

### Формат выходных данных

Выведите  $t$  строк — если  $i$ -й вариант клипа осуществим, выведите в  $i$ -й строке «YES» (без кавычек), иначе выведите «NO» (без кавычек).

### Пример

стандартный ввод	стандартный вывод
3	YES
7 7	NO
3 8	YES
5 6	