

## Задача А. Очень состоятельный крот

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

- Чем хотите пока заняться, состоятельные кроты?
- А что если нам посчитать?
- И то дело!

Один очень состоятельный крот знает только целые неотрицательные числа. Он умеет выполнять на своих счётах три действия:

- 1) получать из числа  $x$  число  $2x$ ;
- 2) получать из числа  $x$  число  $2x+1$ ;
- 3) получать из числа  $x$  число  $\lfloor x/2 \rfloor$  (целую часть при делении  $x$  пополам).

Каждое утро крот выбирает очередную пару целых чисел  $x$  и  $y$ , таких что  $1 \leq x < y \leq 2^l - 1$ , и в течение дня получает из числа  $x$  число  $y$ , выполняя на своих счётах некоторую последовательность действий. Крот хорошо умеет считать, поэтому всегда обходится самой короткой последовательностью действий, достаточной для получения числа  $y$  из числа  $x$ .

Сколько в общей сложности действий на счётах выполнит крот, пока не переберёт все пары чисел в диапазоне от 1 до  $2^l - 1$ ?

### Формат входного файла

В единственной строке записано целое число  $l$  ( $2 \leq l \leq 10^{18}$ ).

### Формат выходного файла

Найдите количество действий на счётах, которое должен будет выполнить крот, и выведите остаток при делении этого числа на  $10^9 + 7$ .

### Пример

<i>standard input</i>	<i>standard output</i>
2	4

В первый день крот получит из числа 1 число 2, выполнив первое действие. Во второй день он получит из числа 1 число 3, выполнив второе действие. В третий день он получит из числа 2 число 3, выполнив сначала третье, а затем второе действие.

## Задача В. Мирный атом

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

*Самое страшное слово в ядерной физике — «Упс!..»*

Белоярская атомная электростанция оказалась сильно повреждена сеишбальным землетрясением, недавно происшедшим в Уральских горах. Ядерный реактор уцелел, но почти все системы управления вышли из строя. В частности, не работают датчики, поэтому персонал может лишь догадываться о том, что происходит внутри реактора. Реактор нужно остановить как можно быстрее! Но для этого необходимо знать текущее положение стержней управления и защиты.

Стержни управления и защиты — это основной инструмент контроля над реактором. Внутри стальной оболочки они содержат сердечник из бора, который хорошо улавливает нейтроны. Когда стержни введены в активную зону реактора, они поглощают нейтроны и тем самым замедляют реакцию. И наоборот, когда они выведены, нейтронов становится больше и реакция ускоряется. На БАЭС стержни могут перемещаться от нулевого положения, когда они полностью выведены, до глубины в  $n$  мм, когда они полностью погружены. Просто погрузить стержни целиком нельзя — если не знать состояния реактора на этот момент, это может привести к плачевным последствиям. Так, именно после погружения стержней на Чернобыльской АЭС в 1986 году реактор стал полностью неуправляемым. Поэтому нужно узнать текущее положение стержней до того, как принимать меры.

В распоряжении у персонала есть лог всех изменений положения стержней. К сожалению, в лог записано лишь то, на сколько миллиметров перемещались стержни, но не сказано, в какую сторону — вверх или вниз. Теперь сотрудникам станции не обойтись без вашей помощи.

### Формат входного файла

В первой строке записаны целые числа  $n$ ,  $s$  и  $k$  — максимальная глубина погружения стержней, исходная глубина погружения и количество записей в логе соответственно ( $0 \leq s \leq n \leq 10^9$ ;  $n > 0$ ;  $1 \leq k \leq 40$ ). Во второй строке записаны  $k$  целых положительных чисел — величины сдвига стержней в хронологическом порядке. Гарантируется, что существует последовательность сдвигов, согласующаяся с логом, при выполнении которой стержни всегда были погружены не менее чем на 0 мм и не более чем на  $n$  мм.

### Формат выходного файла

Выведите два целых числа — минимальную и максимальную возможную глубину погружения стержней в конечный момент времени.

### Пример

<i>standard input</i>	<i>standard output</i>
10 3 3 4 5 1	1 3

## Задача С. Посиделки у Дена

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

*Реклама пива и напитков, изготавливаемых на его основе, не должна использовать образы людей и животных.*

Федеральный закон «О рекламе», гл. 3 ст. 22

Ден позвал в гости  $n$  друзей. Друзья обрадовались и поехали в Ашан за соком.  $i$ -й друг купил  $a_i$  литров сока. После этого друзья приехали к Дену и сели пить сок. Все, включая Дена, выпили сока поровну.

Когда сок закончился, Миша стал жаловаться на то, что он купил его больше, чем выпил сам. Чтобы Миша не расстраивался, Ден достал из кармана сто рублей и дал Мише в качестве компенсации за сок, который тот купил, но не выпил сам. Тогда ещё несколько друзей Дена начали жаловаться на то же, что и Миша. Денег у Дена больше не было, поэтому он предложил всем друзьям, которые купили сока больше, чем выпили сами, справедливо поделить его сто рублей между собой. Как друзья Дена должны поделить эти деньги? Следует считать, что Ден оплатил весь сок, выпитый не теми людьми, которые его покупали, а не только сок, выпитый им самим.

### Формат входного файла

В первой строке записано целое число  $n$  — количество друзей Дена ( $2 \leq n \leq 100$ ). Во второй строке записаны целые числа  $a_1, \dots, a_n$  ( $0 \leq a_i \leq 100$ ). Сумма всех  $a_i$  строго положительна.

### Формат выходного файла

Выведите, сколько рублей из ста должен забрать себе  $i$ -й друг. Округлите все числа вниз до целого количества рублей.

### Примеры

<i>standard input</i>	<i>standard output</i>
3 10 10 10	33 33 33
2 10 0	100 0

## Задача D. Остров Меркет

Имя входного файла:	<i>standard input</i>
Имя выходного файла:	<i>standard output</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Меркет — необитаемый остров в Балтийском море. Исторически западная часть острова принадлежит Швеции, а восточная часть — Финляндии. В 1885 году финны построили на острове маяк. Точных карт острова тогда ещё не существовало, и впоследствии шведская сторона заявила, что маяк находится на её территории. Чтобы урегулировать территориальный спор, было решено провести по острову границу между Швецией и Финляндией так, чтобы площади шведской и финской частей острова совпадали, а маяк целиком находился на территории Финляндии.

Будем считать остров Меркет прямоугольником на плоскости, а маяк — квадратом, расположенным строго внутри этого прямоугольника. Стороны прямоугольника и квадрата параллельны осям координат, юго-западный угол острова имеет координаты  $(0, 0)$ , а северо-восточный —  $(w, h)$ . Граница должна являться ломаной с вершинами в точках с целыми координатами, проходящей от северного побережья острова к южному. Отрезки ломаной должны быть параллельны осям координат. Граница не должна иметь общих точек с западным и восточным побережьями острова и должна иметь ровно по одной общей точке с северным и южным побережьями. Граница не может проходить через маяк, но может касаться его.

Определите, каким образом нужно провести по острову границу между странами.

### Формат входного файла

В первой строке записаны целые числа  $w$  и  $h$  — размеры острова Меркет ( $3 \leq w, h \leq 10^6$ ;  $w \cdot h$  чётно). Во второй строке записаны целые числа  $x$ ,  $y$  и  $d$  — координаты юго-западного угла маяка и длина стороны маяка ( $d > 0$ ;  $0 < x < w-d$ ;  $0 < y < h-d$ ).

### Формат выходного файла

В первой строке выведите количество вершин ломаной, а в следующих строках — целые координаты этих вершин, следуя от вершины на северном побережье к вершине на южном. Вертикальные и горизонтальные звенья ломаной должны чередоваться, первое и последнее звено должны быть вертикальными. Все звенья должны иметь ненулевую длину. Ломаная не должна иметь самопересечений и самокасаний. Количество вершин в ломаной не должно превосходить 1 000. Если границу можно провести несколькими способами, выведите любой из них. Если провести границу с соблюдением всех описанных условий нельзя, выведите «Impossible».

### Примеры

<i>standard input</i>	<i>standard output</i>
6 3 1 1 1	6 4 3 4 2 1 2 1 1 4 1 4 0
10 10 1 1 8	Impossible

## Задача Е. Стихотворный размер

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

*Не мог он ямба от хорея,  
Как мы ни бились, отличить.*  
А. С. Пушкин

В стихосложении *стопой* называется группа из нескольких слогов, один из которых является ударным. Вид стопы определяется количеством слогов и позицией ударного слога в ней.

Обычно в стихотворении используются стопы только одного вида, и этот вид задаёт *размер* стихотворения. Двухсложные стопы с ударением на первый слог задают *хорей*, с ударением на второй слог — *ямб*. Трёхсложные стопы с ударением на первый, второй и третий слог образуют соответственно *дактиль*, *амфибрахий* и *анapest*. Каждая строка стихотворения состоит из нескольких стоп одного вида, причём последняя может присутствовать в строке в усечённом виде.

Вряд ли в ближайшее время найдётся школьник, которому никогда не приходилось и не придётся определять размер стихотворения на уроке литературы. Вы можете помочь им всем, автоматизировав этот процесс.

### Формат входного файла

В первой строке записано целое число  $n$  ( $2 \leq n \leq 100$ ) — количество строк в стихотворении. Следующие  $n$  строк содержат текст стихотворения. Каждое слово стихотворения состоит из одного или более слогов. Если слогов в слове несколько, они разделяются дефисом. Каждый ударный слог заключён в квадратные скобки. Гарантируется, что в каждой строке стихотворения есть как минимум четыре слога. Мы удалили из текста стихотворения все знаки препинания, поэтому теперь в нём встречаются только строчные и заглавные латинские буквы, пробелы, дефисы и квадратные скобки. Длины всех строк не превосходят 50 символов.

### Формат выходного файла

Выведите название размера, которым написано данное стихотворение, — «*trochee*» (хорей), «*iamb*» (ямб), «*dactyl*» (дактиль), «*amphibrach*» (амфибрахий) или «*anapaest*» (анapest). Если же в стихотворении нарушен размер, выведите «*not a poem*».

### Пример

<i>standard input</i>
4 My [un]-cle [high] i-[deals] ins-[pire] him But [when] past [jo]-king [he] fell [sick] he [real]-ly [forced] one [to] ad-[mire] him and [ne]-ver [played] a [shrew]-der [trick]
<i>standard output</i>
iamb

## Задача F. Нанотехнологии

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Специалисты из лаборатории нанотехнологий Уральского федерального университета занимаются созданием вычислителя, работающего на уровне отдельных атомов. Недавно они узнали, что если правильно расположить на плоскости  $n$  атомов тантала, то можно построить на их основе квантовый компьютер, обладающий недостижимыми ранее характеристиками.

К сожалению, теория описывает только некоторые физические характеристики системы атомов, но не их конфигурацию в целом. Физики должны вычислить, как именно нужно расположить атомы, чтобы добиться желаемого эффекта. Сканирующий туннельный микроскоп, который физики используют для своих опытов, может поместить атом тантала только в точки, координаты которых выражены целым числом нанометров, по модулю не превышающим  $10^6$ .

Чтобы определить координаты атомов, учёные обратились за помощью к вам. Они избавились от лишней физической информации и предоставили вам только квадраты расстояний между всеми парами атомов.

### Формат входного файла

Первая строка содержит единственное целое число  $n$  — общее количество атомов ( $2 \leq n \leq 100$ ). Далее в  $n$  строках записана матрица  $a_{ij}$  размера  $n \times n$ . Число  $a_{ij}$  равно требуемому квадрату расстояния между атомами с номерами  $i$  и  $j$  ( $0 \leq a_{ij} \leq 10^9$ ;  $a_{ij} = a_{ji}$ ;  $a_{ii} = 0$ ). Все  $a_{ij}$  целые.

### Формат выходного файла

Если существует ответ, удовлетворяющий данной матрице квадратов расстояний, выведите  $n$  строк.  $i$ -я строка должна содержать два целых числа, по модулю не превосходящих  $10^6$  — координаты  $i$ -го атома. Если существует несколько возможных ответов, выведите любой из них. Если разместить атомы требуемым образом невозможно, выведите «Impossible».

### Примеры

<i>standard input</i>	<i>standard output</i>
3 0 1 4 1 0 1 4 1 0	0 0 1 0 2 0
3 0 1 5 1 0 1 5 1 0	Impossible
3 0 1 0 1 0 1 0 1 0	5 -5 5 -4 5 -5

## Задача G. Конкурс прогнозов

Имя входного файла:	<i>standard input</i>
Имя выходного файла:	<i>standard output</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

На финале чемпионата мира по программированию вручается четыре комплекта золотых, четыре комплекта серебряных и четыре комплекта бронзовых медалей — таким образом, медали получают команды, занявшие первые двенадцать мест.

До финала чемпионата мира в Орландо оставалось несколько дней. Среди ветеранов спортивного программирования из УрГУ разгорелся спор о том, какие команды имеют наиболее высокие шансы завоевать медали на предстоящем соревновании. Вадик предложил каждому, кто хочет сделать прогноз на результаты финала, внести в общий банк пять долларов и высказать несколько предположений вида «Команда вуза  $X$  завоюет медаль достоинства  $Y$ ». Ветераны, сделавшие наибольшее количество верных предположений, должны были поделить деньги из банка между собой.

Но ветераны отказались, заявив, что результаты финала всегда оказываются неожиданными, а терять пять долларов никто не хочет. Тогда Вадик изменил условия конкурса прогнозов. Теперь за то, чтобы сделать свой прогноз, не нужно было платить деньги, но каждому, сделавшему наибольшее количество верных предположений, Вадик платил по пять долларов сам. На таких условиях ветераны согласились участвовать в конкурсе, и каждый из них высказал некоторое количество предположений о том, какая команда какую медаль завоюет.

Зная результаты финала чемпионата мира и все сделанные прогнозы, определите, сколько денег Вадиду пришлось заплатить победителям конкурса.

### Формат входного файла

Первые двенадцать строк содержат названия вузов, завоевавших медали чемпионата мира. Вузы перечислены в порядке занятых мест — с первого по двенадцатое.

В следующей строке записано целое число  $n$  — количество участников конкурса прогнозов ( $1 \leq n \leq 100$ ). Далее следуют  $n$  блоков, содержащих прогнозы ветеранов.  $i$ -й блок начинается с целого числа  $k_i$  — количества предположений, сделанных  $i$ -м участником конкурса. В следующих  $k_i$  строках записаны эти предположения в виде <название\_вуза> : <медаль>. <медаль> — слово «gold», «silver» или «bronze». Предположения каждого участника конкурса непротиворечивы — ни один вуз в них не может завоевать более одной медали, а общее число медалей каждого достоинства не превышает четырёх.

Названия вузов могут содержать строчные и заглавные латинские буквы, символы подчёркивания, точки и запяты. Длина каждого названия составляет не менее одного и не более тридцати символов.

### Формат выходного файла

Выведите, сколько денег должен заплатить Вадик по итогам конкурса прогнозов. Если несколько человек сделало наибольшее количество верных предположений, Вадик выплачивает по пять долларов каждому из них.

## Пример

<i>standard input</i>	<i>standard output</i>
Zhejiang_U U_of_Michigan_at_Ann_Arbor Tsinghua_U St._Petersburg_SU Nizhny_Novgorod_SU Saratov_SU Friedrich_Alexander_U Donetsk_National_U Jagiellonian_U_in_Krakow Moscow_SU Ural_SU U_of_Waterloo 3 6 Moscow_SU : gold St._Petersburg_SU_of_ITMO : gold Warsaw_U : gold Tsinghua_U : gold Nizhny_Novgorod_SU : silver Saratov_SU : silver 6 Warsaw_U : gold Saratov_SU : gold Tsinghua_U : gold Donetsk_National_U : silver St._Petersburg_SU_of_ITMO : silver Ural_SU : bronze 6 Zhejiang_U : gold Tsinghua_U : gold Shanghai_Jiao_Tong_U : gold Fudan_U : gold Moscow_SU : silver Nizhny_Novgorod_SU : silver	15



## Задача Н. Новогодний круиз

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

*И немедленно вытмл...*

В жизни русского человека всегда есть место празднику. Особенно когда несколько поводов для празднования следуют один за другим — Новый год, Рождество, старый Новый год. . . Все отмечают эти праздники по-разному. Кому-то нравится побыть в кругу семьи или друзей, а кого-то, наоборот, тянет в путешествия. В компании «Сибирские железные дороги» решили совместить эти два варианта и запустили специальный круизный поезд «Бодун» по маршруту Владивосток–Москва–Владивосток. Этот поезд будет курсировать всего один раз в год — в новогодние праздники. В поезде будут созданы все условия для того, чтобы любой из праздников можно было отмечать вплоть до следующего. Ну, или до прибытия поезда в пункт назначения.

Спрос на билеты на данный поезд превзошёл все ожидания компании. Их продажу пришлось прекратить уже на третий день, поскольку иначе поезд, вмещающий всех желающих, не встал бы ни на один вокзал. Работники Сибирских железных дорог посчитали, сколько билетов продано на проезд между каждой парой станций, и теперь хотят определить, сколько вагонов должно быть в поезде, чтобы хватило мест для всех пассажиров.

### Формат входного файла

В первой строке записано единственное целое число  $n$  — количество станций, на которых останавливается поезд при следовании в одну сторону ( $2 \leq n \leq 100$ ). Станции пронумерованы числами от 1 до  $n$  в порядке следования поезда, Владивосток имеет номер один, Москва — номер  $n$ . Далее в  $n$  строках записана матрица  $a_{ij}$  размера  $n \times n$ . Число  $a_{ij}$  равно количеству людей, купивших билет от станции  $i$  до станции  $j$  ( $0 \leq a_{ij} \leq 1\,000$ ;  $a_{ii} = 0$ ). Все  $a_{ij}$  целые. Если  $i < j$ , то пассажиры, купившие билет от станции  $i$  до станции  $j$ , сядут на поезд, следующий из Владивостока в Москву, а если  $i > j$ , то на поезд, возвращающийся из Москвы во Владивосток.

### Формат выходного файла

Выведите единственное целое число — минимальное количество вагонов, которое должно быть в поезде, чтобы всем пассажирам хватило мест. В каждом вагоне ровно 36 мест.

### Пример

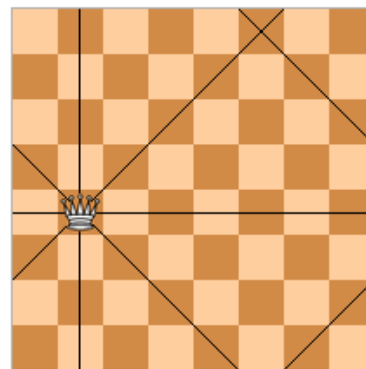
<i>standard input</i>	<i>standard output</i>
3 0 180 180 0 0 180 360 0 0	10

## Задача I. Зиниум 2

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Напомним, что Зиниум — это шахматная доска размером  $n \times n$  клеток. Клетка в левом нижнем её углу имеет координаты  $(1, 1)$ , а клетка в правом верхнем углу — координаты  $(n, n)$ . По легенде, если на доске расставить  $n$  ферзей таким образом, что ни один из них не будет атаковать другого, то освобождённая энергия Зиниума изменит мир до неузнаваемости. Реки повернутся вспять, небо упадёт на землю, люди научатся называть вещи своими именами. . .

Все верили в легенду про Зиниум, пока артефакт не попал в руки к Игорю. Когда Игорю удалось расставить  $n$  ферзей требуемым образом, ничего не произошло (во всяком случае, Игорь не заметил ничего необычного). Тогда Игорь предположил, что в легенде говорилось не об обычной, а о торической шахматной доске. Чтобы получить торическую шахматную доску размера  $n \times n$ , нужно взять обычную доску такого же размера, после чего склеить её верхнюю горизонталь с нижней, а левую вертикаль с правой. На рисунке показано, какие клетки торической шахматной доски  $8 \times 8$  держит под боем один ферзь. Чтобы проверить свою гипотезу, Игорь пытается расставить на торической доске  $n$  ферзей так, чтобы ни один из них не атаковал другого. Помогите ему в этом.



### Формат входного файла

В единственной строке записано целое число  $n$  ( $4 \leq n \leq 10^5$ ).

### Формат выходного файла

Если искомая расстановка существует, выведите в первой строке «Yes», а во второй —  $n$  целых чисел.  $i$ -е число должно равняться  $y$ -координате ферзя,  $x$ -координата которого равна  $i$ . Если возможных расстановок несколько, выведите любую из них. Если расстановки не существует, в единственной строке выведите «No».

### Примеры

<i>standard input</i>	<i>standard output</i>
5	Yes 2 4 1 3 5
8	No

## Задача J. Сейсмические волны

Имя входного файла:	<i>standard input</i>
Имя выходного файла:	<i>standard output</i>
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

*In case of fire, exit building before  
tweeting about it.*

Сейчас уже никого не увидишь историями о том, как человек написал в блог или социальную сеть о чрезвычайном происшествии до того, как вызвать службу спасения. Японские власти, воспользовавшись такой привязанностью граждан к интернет-технологиям, предложили им использовать Twitter, чтобы предупреждать своих друзей и близких о землетрясении. Действительно, скорость распространения сейсмических волн составляет не более пяти километров в секунду, поэтому люди, находящиеся далеко от эпицентра, смогут прочитать о нём за несколько секунд до прихода сейсмической волны. Для кого-то эти секунды могут стать решающими. . .

У каждого пользователя Twitter есть список *фолловеров*. *Твит* (сообщение) пользователя видят все его фолловеры. Каждый из них может сделать *ретвит* этого твита, то есть повторить полученный твит, чтобы он стал доступен уже их фолловерам. Эти фолловеры могут, в свою очередь, могут снова сделать ретвит, и так далее. . . Каждый раз при ретвите к сообщению автоматически спереди дописывается строка «RT @nick: », где *nick* — имя пользователя, чей твит (или ретвит) ретвитится. Длина любого сообщения не может превышать 140 символов, поэтому, если ретвит превышает это ограничение, он не отправляется вовсе.

Для проведения эксперимента МЧС Японии выбрало несколько человек и решило вычислить, скольких из них можно предупредить, сообщив о надвигающемся землетрясении лишь одному.

### Формат входного файла

В первой строке дано целое число  $n$  — количество пользователей Twitter, выбранных для эксперимента ( $2 \leq n \leq 100$ ). Далее в  $n$  строках следуют их описания в следующем формате: имя пользователя, количество его фолловеров и список их имён. Количество фолловеров — целое неотрицательное число. В списке содержатся только имена других выбранных для эксперимента пользователей, имена в списке не повторяются. Все имена — непустые строки, состоящие из строчных и заглавных латинских букв, длиной не более 30 символов. Имена всех пользователей различны. В последней строке содержится твит о землетрясении, отправленный первым из описанных пользователей — непустая строка, состоящая из символов с ASCII кодами от 32 до 127, длиной не более 140 символов.

### Формат выходного файла

В первой строке выведите количество пользователей, которые узнают о землетрясении. Затем выведите их имена в произвольном порядке, по одному в строке.

### Пример

<i>standard input</i>	<i>standard output</i>
5 Miyako 1 Sae Yuno 2 Miyako Sae Nazuna 1 Nori Sae 1 Yuno Nori 1 Nazuna Hidamari is going to collapse!!!	3 Miyako Sae Yuno

## Задача К. Просторный офис

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Программисты компании «Периметр» работают над  $n$  проектами. Начальник программистов Шифтмэн прекрасно понимает, насколько важны комфортные условия для продуктивной работы — в компании нет ни дресс-кода, ни фиксированного рабочего графика, зато на кухне всегда есть чай и свежие киви. А когда команда проекта «Диплодок» стала жаловаться на то, что после выхода на работу новых сотрудников в комнате не протолкнуться, Шифтмэн понял — пора искать более просторный офис.

Новое офисное здание нашлось быстро. Оно расположено недалеко от метро, возле уютного парка. Кроме того, в подвале здания много парковочных мест. Узнав, что в новом офисе  $n$  больших комнат, Шифтмэн решил выделить по комнате каждому проекту, чтобы сотрудники создали там уникальную для каждого проекта рабочую атмосферу. Менеджер каждого проекта знает, какая комната идеально подойдёт для его команды — с одной стороны, комната должна быть не слишком тесной, а с другой — не слишком большой, чтобы сотрудники не боялись того, что к ним могут подселить новый проект. Помогите менеджерам поделить комнаты самим, быстро и без вмешательства начальства.

### Формат входного файла

В первой строке записано целое число  $n$  — количество проектов в Периметре ( $2 \leq n \leq 1000$ ). Во второй строке записаны  $n$  целых чисел — площади всех комнат в новом офисе. В  $i$ -й из следующих  $n$  строк записаны два целых числа — минимальная и максимальная площади комнаты, в которой согласна сидеть команда  $i$ -го проекта (естественно, минимальная площадь не превосходит максимальной). Все указанные площади положительные и не превосходят 10 000.

### Формат выходного файла

Если существует единственный способ рассадить команды по комнатам так, чтобы все команды остались довольны, в первой строке выведите «Perfect!», а во второй строке — перестановку чисел от 1 до  $n$ .  $i$ -е число должно обозначать номер комнаты, которую должна занять команда  $i$ -го проекта. Комнаты занумерованы числами от 1 до  $n$  в том порядке, в котором они описаны во входных данных. Если возможных вариантов рассадки несколько, выведите «Ask Shiftman for help.» Если рассадить команды требуемым образом нельзя, выведите «Let's search for another office.»

### Примеры

<i>standard input</i>	<i>standard output</i>
3 40 50 60 30 70 20 40 60 60	Perfect! 2 1 3
3 40 50 70 30 70 20 50 60 60	Let's search for another office.