

## Задача А. АВВА или ВАВА?

Имя входного файла: `abba.in`  
Имя выходного файла: `abba.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

У одного из меломанов, участвующих в подготовке этой олимпиады, любимой группой так и остаётся АВВА (да простят его некоторые участники, предпочитающие что-нибудь «потяжелее»). Соответственно, когда речь заходит о примерах различных алгоритмов на строках, он в первую очередь составляет строки из двух букв  $a$  и  $b$ .

Собственно из анализа таких примеров и родилась следующая задача. Пусть мы хотим прочитать в строке буквосочетание  $ab$ . При этом  $a$  и  $b$  не обязательно должны стоять подряд, достаточно, чтобы  $a$  встречалось в строке раньше, чем  $b$ . Как составить строку минимально возможной длины, чтобы это буквосочетание можно было прочитать *ровно*  $K$  способами? Например, в строке  $abba$  это буквосочетание можно прочитать дважды, но эта строка не будет самой короткой для  $K = 2$ , а в строке  $abab$  его можно прочитать три раза, и более короткую строку для  $K = 3$  составить нельзя.

Напишите программу, которая и будет составлять подобную строку минимальной длины для заданного натурального  $K$  или будет определять, что сделать это невозможно.

### Формат входного файла

Входные данные содержат только одно натуральное число  $K$  ( $1 \leq K \leq 10^9$ ).

### Формат выходного файла

Выведите строку, удовлетворяющую условию задачи, или слово «Impossible», если искомую строку составить невозможно. Если искомым строк минимальной длины несколько, то выведите любую из них.

### Примеры

	<code>abba.in</code>	<code>abba.out</code>
1		<code>ab</code>
3		<code>abab</code>

## Задача В. Сортировка

Имя входного файла: `sort.in`  
Имя выходного файла: `sort.out`  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 64 мегабайта

Учительница по программированию задала Вовочке задачу — отсортировать массив из  $N$  чисел от 1 до  $N$  по возрастанию :). Вовочка поступает так: он просматривает массив слева направо, и, если замечает два элемента, стоящих рядом, таких, что правый меньше левого, он меняет их местами. Так он поступает, пока массив не будет отсортирован. Но Вовочка — очень ленивый ученик. В какой-то момент ему надоело сортировать числа, и он решил посчитать, сколько ещё описанных выше обменов нужно сделать. Помогите ему.

### Формат входного файла

В первой строке входного файла находится натуральное число  $N$  ( $1 \leq N \leq 1\,000\,000$ ). Во второй строке через пробел находятся  $N$  различных целых чисел, каждое из которых не меньше 1 и не больше  $N$ .

### Формат выходного файла

Выведите одно число — искомое количество обменов. Так как ответ может быть очень большим и сложным, выведите его по очень простому модулю 2.

### Примеры

<code>sort.in</code>	<code>sort.out</code>
5 1 2 5 4 3	1

## Задача С. RPQ (Range Permutation Query)

Имя входного файла: `rpq.in`  
Имя выходного файла: `rpq.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

От некоторых школ в командной олимпиаде по информатике участвует очень много команд. Учитель одной из таких школ раздал для регистрации своим командам номера: 1, 2, 3 и т. д. Для того чтобы проверить, все ли команды зарегистрировались, учитель выписал из таблицы регистрации только номера команд своей школы, но в том порядке, в котором команды регистрировались.

После нелёгких подсчётов оказалось, что зарегистрировались все. Но в процессе решения этой задачи учитель сформулировал следующую: сколькими способами можно выбрать стоящие подряд в этом списке  $K$  номеров команд, чтобы они образовывали некоторую перестановку чисел от 1 до  $K$ ? Например, если от школы участвуют всего три команды, то при порядке регистрации 3 1 2 таких способов будет три (для  $K = 1, 2, 3$ ), а при регистрации в порядке 2 3 1 — всего два (для  $K = 1$  и  $K = 3$ ).

### Формат входного файла

В первой строке входных данных находится одно число  $N$  ( $1 \leq N \leq 1\,000\,000$ ) — количество команд, участвующих в олимпиаде от этой школы. Во второй строке находятся  $N$  натуральных чисел от 1 до  $N$  в том порядке, в котором команды регистрировались на олимпиаду.

Гарантируется, что каждое число встречается в этой строке ровно один раз.

### Формат выходного файла

Выведите одно число, обозначающее число способов выбрать из списка несколько стоящих подряд команд, удовлетворяющих условию задачи.

### Примеры

<code>rpq.in</code>	<code>rpq.out</code>
3 2 3 1	2

## Задача D. Экономия на рейсах

Имя входного файла:	dist.in
Имя выходного файла:	dist.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Как вы помните, Джонни работает в министерстве финансов одной небольшой страны. По роду службы ему приходится инспектировать отечественные авиакомпании и изучать маршруты, которые те предлагают.

В стране есть  $N$  городов. По государственным законам авиакомпания должна предоставлять услуги двустороннего перелёта между некоторыми парами городов, причём в целях стандартизации продолжительность любого полёта должна составлять некоторое целое число часов. Также для фиксированной пары городов длительность рейса не должна зависеть от направления перелёта.

Авиакомпания называется *крупной*, если с помощью рейсов этой компании можно добраться из любого города страны до любого другого. *Крупная* авиакомпания называется *экономной*, если при этом количество маршрутов, ею предлагаемое, минимально возможное, которое может быть у *крупной* компании.

Государственная антимонопольная служба возбудила расследование против *крупной* авиакомпании «Aero-float». Ей предъявлено обвинение в излишней неэкономности. Джонни было поручено проинспектировать «Aero-float» в целях обнаружения финансовых махинаций, но авиакомпания отказалась раскрывать полный список выполняемых ею прямых рейсов. После продолжительных переговоров руководство компании согласилось сообщить Джонни информацию о минимально возможной продолжительности перелёта между каждой парой городов, если использовать только прямые рейсы «Aero-float» и не учитывать время, затрачиваемое на пересадки.

Используя эту информацию, помогите Джонни установить: может ли компания «Aero-float» быть *экономной* или нет? Более формально, установите, существует ли набор рейсов, при котором длины кратчайших маршрутов именно такие, как было сообщено Джонни, и при котором компания является *экономной*?

### Формат входного файла

Ваша программа должна будет обработать несколько наборов входных данных. В первой строке входного файла идёт их количество  $C$  ( $1 \leq C \leq 1500$ ). Далее идут  $C$  блоков входных данных в следующем формате.

В первой строке блока находится единственное целое число  $N$  ( $2 \leq N \leq 1500$ ) — количество городов в стране.

Далее идут  $N$  строк по  $N$  целых чисел  $T_{i,j}$  ( $1 \leq T_{i,j} \leq 1\,000\,000$ ):  $j$ -е число в  $i$ -й строке равняется минимальному времени в часах, требуемому на перемещение из  $i$ -го города в  $j$ -й без учёта времени, затрачиваемого на пересадки.

Гарантируется, что предоставленная информация корректна, то есть существует некоторый набор рейсов, который соответствует данному набору чисел  $T_{i,j}$ .

Сумма квадратов  $N$  по всем наборам входных данных не превосходит 2 250 000.

### Формат выходного файла

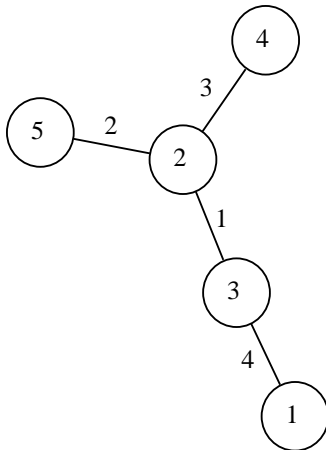
Для каждого набора входных данных выведите «YES», если компания «Aero-float» может являться *экономной*, или «NO» в противном случае.

## Примеры

dist.in	dist.out
2	YES
5	NO
0 5 4 8 7	
5 0 1 3 2	
4 1 0 4 3	
8 3 4 0 5	
7 2 3 5 0	
3	
0 2 2	
2 0 2	
2 2 0	

## Примечание

Для первого набора ниже приведён возможный вариант предлагаемых компанией маршрутов, при котором она является *экономной*. На рисунке отрезками соединены те пары городов, между которыми есть прямые рейсы, над каждым отрезком указана продолжительность соответствующего рейса.



## Задача Е. Поливальная машина

Имя входного файла: `water.in`  
Имя выходного файла: `water.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Город  $N$ , в отличие от города  $M$ , расположен на склоне одного холма. Чистоту улиц этого города обеспечивает единственная поливальная машина. Бензин нынче дорог, поэтому движение муниципального транспорта «в гору» признано слишком расточительным.

Карта города представляет собой прямоугольник, разбитый на  $H \times W$  клеток, где  $H$  и  $W$  — высота и ширина карты в клетках соответственно. Часть клеток заняты зданиями, остальные соответствуют улицам и площадям, которые и требуется помыть.

Поливальная машина начинает свой путь в одной из клеток самого верхнего ряда, не занятой зданиями. Она может полить асфальт в текущей клетке и переместиться в любую из двух соседних клеток этого же ряда, не занятых зданиями. Объехать здания, не поднимаясь при этом в гору, невозможно. Поливальная машина также может переместиться в соседнюю по вертикали свободную клетку из нижнего ряда.

Помогите узнать экономному муниципалитету, какое максимальное количество свободных клеток поливальная машина сможет помыть, не поднимаясь при этом в гору? Определите также, какое минимально возможное расстояние ей при этом придётся преодолеть. Считается, что, перемещаясь в соседнюю по горизонтали или вертикали клетку, поливальная машина преодолевает одну единицу расстояния.

### Формат входного файла

В первой строке входного файла находятся натуральные числа  $H$  и  $W$  — высота и ширина карты города ( $1 \leq W \leq 1000$ ,  $1 \leq H \leq 1000$ ). В следующей строке расположено единственное натуральное число  $K$  — номер столбца клетки первой строки, в которой поливальная машина начинает движение ( $1 \leq K \leq W$ ). Гарантируется, что эта клетка свободна от зданий.

Каждая из следующих  $H$  строк содержит  $W$  символов '#' и '.', означающих, соответственно, клетки со зданиями и без.

### Формат выходного файла

Выведите два целых числа — максимальную площадь в клетках, которую может обработать поливальная машина, и минимальное расстояние, которое ей для этого придётся преодолеть.

### Примеры

<code>water.in</code>	<code>water.out</code>
8 8 7 ..#...# ##...# ...#..# .##..#. ..#.#... #.#.#... #.#...## ##.####	21 23

## Задача F. Фотооптимизация

Имя входного файла: `photo.in`  
Имя выходного файла: `photo.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

На Международной олимпиаде по информатике некоторые участники, конечно же, получают удовольствие именно от решения предложенных задач, но большинство — от полученных в новой стране впечатлений. Впечатления принято запечатлевать на фотоаппарат. Участник  $T$  решил подойти к процессу съёмки с научной (по его мнению) точки зрения. Он находится на круглой обзорной площадке и желает заснять сразу два интересных объекта, местоположение каждого из которых на земле мы будем описывать с помощью отрезка.  $T$  выбирает точку для съёмки внутри этой площадки так, чтобы суммарная площадь двух треугольников, образованных концами соответствующих отрезков и выбранной точкой на обзорной площадке была как можно больше.

Помогите  $T$  с выбором такой точки. Возможность заснять сразу два объекта при этом анализировать не нужно, мы лишь действуем в рамках модели, сформулированной  $T$ . Если объекты загораживают друг друга, то этим также нужно пренебречь и считать площади независимо.

### Формат входного файла

В первой строке входного файла находятся число  $T$  — количество тестов ( $1 \leq T \leq 10^5$ ). Далее следуют описания тестов. В первой строке каждого описания содержится 4 числа  $x_1, y_1, x_2, y_2$ , характеризующие координаты концов первого отрезка. Во второй строке —  $x_3, y_3, x_4, y_4$ , описывающие второй отрезок. В третьей строке записаны три числа  $x_0, y_0, R$  — координаты центра круговой площадки и её радиус. Все числа целые, по модулю не превосходящие 1000. Радиус положителен.

Гарантируется, что отрезки невырождены и не пересекаются с круговой площадкой. Также гарантируется, что они не имеют общих внутренних точек.

### Формат выходного файла

Для каждого теста выведите координаты точки внутри круга, удовлетворяющей условию задачи. Если таких точек несколько, то выведите любую из них. Координаты следует выводить с как можно большим числом знаков после десятичной точки. Соответствующая сумма площадей должна отличаться от правильного ответа по абсолютной или относительной величине не больше чем на  $10^{-6}$ .

### Примеры

<code>photo.in</code>	<code>photo.out</code>
1 0 0 1 0 0 0 -1 0 0 2 1	0.0000000000 3.0000000000

## Задача G. И последние станут первыми

Имя входного файла: `frozen.in`  
Имя выходного файла: `frozen.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Командная олимпиада, о которой идёт речь в этой задаче, проходила по тем же правилам, что и олимпиада, в которой вы сейчас участвуете. Возможно, за одним исключением: на ней было введено ограничение на суммарное число посылок от одной команды, которые можно было потратить как на одну задачу, так и распределить их по всем задачам некоторым образом.

Как и сегодня, за час до окончания олимпиады таблица результатов была «заморожена». Каково же было удивление участников, когда при подведении итогов выяснилось, что в окончательной таблице все команды расположились строго в обратном порядке (по отношению к порядку, зафиксированному «заморозкой»). Так как в окончательной таблице результаты участников не отражены, вам предлагается определить, могло ли такое быть в принципе, или это результат технического сбоя системы.

### Формат входного файла

Первая строка содержит три натуральных числа:  $N$  — количество команд,  $K$  — количество задач,  $L$  — лимит на суммарное число посылок по задачам ( $N \leq 1000$ ,  $K \leq 12$ ,  $K \leq L \leq 200$ ).

Далее следуют  $N$  строк, описывающих саму таблицу на момент заморозки.  $i$ -я строка содержит название команды (строка из латинских символов и цифр длины не более 50) и  $K$  элементов, показывающих успех команды по каждой из задач.  $j$ -й элемент таблицы имеет вид  $*Wrong(Time)$ , где:

\* — знак ‘+’ или ‘-’, показывающий, решила ли команда задачу;

$Wrong$  — количество неуспешных посылок по данной задаче;

$Time$  — минута, на которой команда сдала задачу ( $0 \leq Time \leq 239$ ); если данная задача ещё не была решена,  $Time = 0$ .

Команды упорядочены от первого места на момент заморозки к последнему. Напоминаем, что команды сортируются по числу решённых задач, а при равном количестве — по увеличению штрафного времени, которое складывается из времени сдачи задачи и общего количества неудачных попыток по решённым задачам, умноженного на 20. Продолжительность олимпиады — 5 часов.

Гарантируется, что ни одна команда не превысила лимит на число посылок и что никакие две команды не имеют одинаковый с точки зрения сортировки результат.

### Формат выходного файла

Если порядок команд теоретически мог измениться за последний час соревнований на обратный, то сначала выведите строку «Possible», а затем подробную таблицу окончательных результатов в формате, аналогичном таблице из входного файла (см. пример). Если соревнование так закончиться не могло, то выведите единственную строку «Impossible».

### Примеры

<code>frozen.in</code>	<code>frozen.out</code>
<code>3 3 10 winner +2(235) +3(170) -2(0) middle -2(0) +1(30) -0(0) looser -1(0) +5(30) -0(0)</code>	<code>Possible looser +1(240) +5(30) +0(240) middle +5(241) +1(30) +0(240) winner +2(235) +3(170) +2(240)</code>
<code>2 2 20 winner +10(1) +0(10) looser -0(0) +1(1)</code>	<code>Impossible</code>



## Задача Н. Бег по кругу

Имя входного файла: `run.in`  
Имя выходного файла: `run.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Спортивный программист для достижения вершин своего мастерства должен быть натренирован в совершенно разных аспектах, в том числе и физически. Кто-то для этого садится на велосипед, кто-то ныряет в бассейн, а молодой программист Влад бегают по стадиону. Но из-за неаккуратного обращения с личными вещами его секундомер может измерять время только в минутах, без указания секунд и тем более их долей.

Чтобы следить за прогрессом своего ученика, тренеру Влада приходится довольствоваться показаниями этого прибора. Каждый раз, когда Влад пробегает мимо тренера, сделав очередной круг по стадиону, тот записывает в блокнот показания секундомера в минутах. Фактически показания секундомера соответствуют целому числу минут, прошедших к определенному моменту времени. Причём, если секундомер показывает, например, 1, то это может обозначать и время ровно 2 минуты, так как  $1.(9) = 2$ .

На контрольной тренировке Влад бежал с постоянной скоростью, однако по записям тренера не так легко сказать, с какой именно. Кроме того, секундомер был, возможно, запущен до того как Влад начал бегать. Напишите программу, которая поможет тренеру определить за какое минимальное, а также максимальное возможное время Влад мог пробегать каждый круг.

### Формат входного файла

В первой строке входного файла находится единственное натуральное число  $N$  — количество записей в блокноте тренера ( $2 \leq N \leq 10^5$ ). В следующей строке находятся сами эти записи — разделённые пробелами целые числа  $a_1, a_2, \dots, a_N$  ( $0 \leq a_1 \leq a_2 \leq \dots \leq a_N \leq 10^6$ ). Здесь  $a_1$  соответствует времени, когда Влад пробежал мимо тренера в первый раз.

### Формат выходного файла

Выведите два неотрицательных вещественных числа, разделённых пробелом, — минимальное и максимальное возможное количество минут, за которое спортсмен пробежал один круг. Ваш ответ должен отличаться от правильного менее чем на  $10^{-3}$ .

Если ответа не существует, то есть Влад не мог бежать с постоянной скоростью так, чтобы записи тренера получились именно такими, в единственной строке выведите «No solution».

### Примеры

<code>run.in</code>	<code>run.out</code>
5 2 3 5 6 8	1.33333 1.66667
5 1 6 9 14 17	4 4
3 1 5 6	No solution
5 1 1 2 3 3	0.5 0.75

### Примечание

Во втором тесте время 4 соответствует показаниям секундомера 1.(9), 6.0, 9.(9), 14.0, 17.(9).

В четвёртом примере минимальное время соответствует показаниям секундомера 1.5, 1.(9), 2.5, 3.0, 3.5, а максимальное — показаниям 1.0, 1.75, 2.5, 3.25, 3.(9).

## Задача I. Календарь

Имя входного файла: `days.in`  
Имя выходного файла: `days.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Как вы помните, Джонни работает в министерстве финансов одной небольшой страны. На этот раз он решил очень тщательно распланировать бюджет. Для этого ему первым делом необходимо выяснить, сколько же будет выходных дней в каждом интересующем его году (без учёта праздников, которые и в этой стране то и дело переносят).

Так как Джонни смотрит в будущее, то его интересуют лишь года, которые ещё не наступили. Но он верит в успехи биоинформатики, касающиеся увеличения продолжительности жизни, и хочет рассчитать всё заранее, поэтому его интересуют и очень отдалённые даты. При этом он предположил, что достаточно рассчитать лишь некоторые из интересующих его годов, чтобы получить общую картину.

### Формат входного файла

В первой строке входного файла содержится целое число  $Q$  — количество лет, которые интересуют Джонни ( $1 \leq Q \leq 1\,000\,000$ ). Далее в  $Q$  строках содержатся номера годов  $Y_i$ , по одному на строке ( $2013 \leq Y_i \leq 10^9$ ).

### Формат выходного файла

Для каждого года в отдельной строке выведите количество выходных дней в соответствующем году.

### Примеры

<code>days.in</code>	<code>days.out</code>
1	104
2013	

### Примечание

Напомним, что в неделе семь дней, выходными считаются суббота и воскресенье. Сегодня четырнадцатое октября две тысячи двенадцатого года, воскресенье. В невисокосных годах 365 дней, в високосных — 366. Год называется високосным, если он делится на 400, или если он делится на 4, но не делится на 100.

## Задача J. Несчастный король

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Максим и Лёша поехали на сборы в Петрозаводск. После тура они решили не ходить на дорешивание, а поиграть в шахматы. Игра проходит на бесконечной доске, и в какой-то момент у Максима осталось только две ладьи и король, а у Алексея — один король. А на бесконечной доске в такой ситуации победить затруднительно. Тогда Максим заменил своего короля на запасённую ещё одну ладью и ситуация изменилась.

Помогите Максиму поставить мат тремя ладьями на бесконечной доске.

### Формат входного файла

Это интерактивная задача. При запуске решения на стандартный поток ввода поступают 8 чисел — координаты короля и трёх ладей на поле. Координаты не превосходят по модулю 100. Гарантируется, что в начальный момент никакие две фигуры не стоят на одной клетке, и король не находится под боем ни одной из ладей. Первый ход делает Максим.

На каждый ход Максима вводится ответный ход Алексея — перемещение короля  $d_x, d_y$  относительно текущей позиции ( $0 \leq |d_x|, |d_y| \leq 1$ ). В случае, если  $|d_x| = |d_y| = 0$ , программа должна немедленно завершиться (это означает, что был поставлен мат, пат или сделан некорректный ход).

### Формат выходного файла

Для каждого хода выводите на стандартный поток вывода три числа — номер ладьи (1, 2 или 3) и перемещение ладьи  $l_x, l_y$  ( $|l_x| + |l_y| > 0; |l_x| \cdot |l_y| = 0$ ). Ход должен быть корректным, т. е. ладья не может пойти в занятую клетку, перепрыгнуть через другую фигуру. Перемещение ладьи не должно быть больше, чем на 1 000 клеток.

### Примеры

stdin	stdout
2 0 1 2 0 4 4 1	1 0 -1
1 0	2 3 0
-1 -1	3 -2 0
0 0	

### Примечание

Вывод должен завершаться переводом строки и сбросом буфера потока вывода. Для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

Программа не должна делать более 40 ходов. Если после хода программы мат не поставлен, а король сделать ход не может, то считается, что тест не пройден.

Ладья может ходить на произвольное количество клеток по горизонтали или по вертикали. Король же может ходить в любую соседнюю клетку, которая не находится под боем ладьи, в том числе, он может «съесть» ладью, если клетка, в которой ладья стоит, не находится под боем другой ладьи. Обратите внимание, что в случае, если король съест ладью, то выигрыш станет невозможным, и тест не будет пройден.

Мат — ситуация в шахматах, когда король находится под ударом ладьи, а игрок не может сделать ни одного хода, чтобы его избежать. Пат — ситуация в шахматах, когда король не находится под ударом ладьи, но при этом игрок не может сделать ни одного хода.

Обратите внимание, что тестирующая система сообщает подробный протокол взаимодействия вашей программы и программы жюри. Гарантируется, что существует программа, которая на всех тестах жюри умеет ставить мат.

Пример ответа тестирующей системы для примера из условия.

Starting position: king on (2 0), rooks on (1 2) (0 4) (4 1)

Turn 1

Max moves 1st rook to (1 1)... OK.

Alex moved king to (3 0)

King on (3 0), rooks on (1 1) (0 4) (4 1)

Turn 2

Max moves 2nd rook to (3 4)... OK.

Alex moved king to (2 0)

King on (2 0), rooks on (1 1) (3 4) (4 1)

Turn 3

Max moves 3rd rook to (2 1)... OK.

It's nowhere for king to go :-)

Game over! Checkmate - Max wins!

Положение фигур по ходу игры.

