

## Задача А. Парикмахер армии магов

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Пётр, избранный лидером новой армии магов, столкнулся с проблемой. У всех магов, призванных в армию, были шикарные бороды, что совершенно неприемлемо для солдат. Поэтому Пётр приказал всем новобранцам как можно быстрее сбрить бороду. Конечно, все маги отказались сделать это, сославшись на то, что им неизвестно ни одно заклинание бритья. К счастью, в Королевстве нашёлся маг-парикмахер Цирюльникус, согласившийся побрить всех новобранцев.

С помощью заклинания «**Fusion Power**» Цирюльникус за одну минуту может сбрить бороду не более чем у  $k$  магов. Для достижения полного эффекта каждый маг должен попасть под действие заклинания два раза: первое заклинание бреет чисто, второе — ещё чище. Пётр назначил каждому новобранцу время, когда тот обязан явиться к Цирюльникусу. Увы, дисциплина в новой армии пока хромает, поэтому каждый маг, хоть и придёт вовремя, но просидит у Цирюльникуса ровно столько, на сколько ему хватит терпения, а затем исчезнет.

Определите, сможет ли Цирюльникус сбрить бороду у всех магов новой армии, прежде чем они исчезнут.

### Формат входного файла

В первой строке через пробел записаны целые числа  $n$  и  $k$  ( $1 \leq n, k \leq 100$ ) — количество новобранцев в армии и количество магов, к которым Цирюльникус может одновременно применить своё заклинание. В  $i$ -й из следующих  $n$  строк через пробел записаны целые числа  $t_i$  и  $s_i$  ( $0 \leq t_i \leq 1000$ ;  $2 \leq s_i \leq 1000$ ) — время в минутах, к которому  $i$ -й маг должен явиться к Цирюльникусу, и время в минутах, которое он готов там провести, включая время бритья.

### Формат выходного файла

Если Цирюльникус сможет сбрить бороду у всех магов, выведите в первой строке «**Yes**», а в  $i$ -й из следующих  $n$  строк выведите пару целых чисел  $p_i, q_i$  — моменты времени, в которые Цирюльникус должен применить заклинание к  $i$ -му магу ( $t_i \leq p_i < q_i \leq t_i + s_i - 1$ ). Если хотя бы один маг исчезнет раньше, чем будет полностью побрит, выведите в единственной строке «**No**».

### Примеры

<code>input.txt</code>	<code>output.txt</code>
3 2 1 3 1 3 1 3	Yes 1 2 1 3 2 3
2 1 1 3 1 3	No

## Задача В. Космический боулинг

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Жители планет, вращающихся вокруг пульсара PSR 2010+15, очень любят играть в космический боулинг. На огромном поле выставляются цилиндрические кегли единичного диаметра. Игрок встаёт в некоторую точку поля и кидает из неё шар с целью уничтожить как можно больше кеглей. Шар после броска катится по прямой, всё время касаясь поля, пока не укатывается за его пределы. Если шар касается кегли, она дематериализуется, а шар не меняет траекторию своего движения. Для того чтобы выбить *страйк*, игрок должен одним броском уничтожить не менее  $k$  кеглей.

К сожалению, инопланетяне ещё не изобрели автомат, который бы возвращал шары, укатившиеся за пределы поля. Вместо этого они используют автомат, который перед каждым броском материализует новый шар из вакуума. Игрок говорит автомату диаметр и через секунду получает шар в точности такого диаметра.

Перед броском инопланетянина Вась-Вася на поле осталось  $n$  кеглей. Помогите Вась-Васю определить минимальный диаметр шара, которым он сможет выбить страйк.

### Формат входного файла

В первой строке через пробел записаны целые числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 200$ ). В  $i$ -й из следующих  $n$  строк через пробел записаны целые числа  $x_i$  и  $y_i$  ( $-10^5 \leq x_i, y_i \leq 10^5$ ) — координаты центров кеглей. Центры никаких двух кеглей не совпадают.

### Формат выходного файла

Выведите минимально возможный диаметр шара, которым можно выбить страйк, с абсолютной или относительной погрешностью не более  $10^{-6}$ . Если страйк можно выбить шаром сколь угодно малого диаметра, выведите «0.000000».

### Пример

input.txt	output.txt
5 4 0 4 0 6 6 4 6 6 3 0	1.0000000000

## Задача С. Праздничный фейерверк

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Денис отвечает за подготовку праздничного фейерверка, посвящённого 90-летию Уральского государственного университета. Он купил  $n$  ракет и начал думать, по какому принципу их нужно запускать. После пары бессонных ночей Денис остановился на следующем алгоритме.

Все  $n$  ракет выстраивают на земле в одну линию. Залпы фейерверка происходят с интервалом в десять секунд. Во время первого залпа запускаются крайняя левая и крайняя правая ракеты. После того, как сделано уже  $i$  залпов, выбираются все непустые отрезки между двумя соседними запущенными ракетами. На каждом таком отрезке случайным образом равновероятно выбирается ракета. Все выбранные ракеты запускаются во время  $(i + 1)$ -го залпа. Алгоритм работает до тех пор, пока все ракеты не будут запущены.

Оцените, сколько в среднем секунд будет продолжаться такой фейерверк.

### Формат входного файла

В единственной строке записано целое число  $n$  ( $3 \leq n \leq 400$ ) — количество ракет, купленных Денисом.

### Формат выходного файла

Выведите математическое ожидание длительности фейерверка в секундах, с абсолютной или относительной погрешностью не более  $10^{-6}$ .

### Пример

<code>input.txt</code>	<code>output.txt</code>
5	26.666666666666

Сначала запустятся ракеты с номерами 1 и 5. Через 10 секунд с вероятностью  $1/3$  запустится ракета номер 3, в этом случае ещё через 10 секунд запустятся ракеты 2 и 4, и салют закончится за 20 секунд. Если же во время второго залпа запустится ракета 2 или ракета 4 (это произойдёт с вероятностью  $2/3$ ), то салют закончится за 30 секунд.

## Задача D. Племя аиндилъяква

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В языке племени австралийских аборигенов аиндилъяква нет числительных. Ни один аиндилъяква не сможет сказать: «Я поймал восемь рыб». Вместо этого он говорит: «Я поймал столько рыб, сколько камней вот в этой куче».

Профессор Брайан Баттеруорт нашёл поляну, на которой лежали три кучи камней, и решил проверить, умеют ли аборигены считать. Профессор отвёл одного из аборигенов на поляну и дал ему задание: показать на две кучи с минимальной разницей количества камней в них и ответить, чему равна эта разница. И абориген справился! Поскольку он не мог выразить разницу словами, он сходил на берег моря и принёс на поляну кучу с таким количеством камней.

Профессор решил повторять свой эксперимент с другими аборигенами, пока какой-то из них не укажет на две кучи с одинаковым количеством камней. Все камни, которые приносят аборигены, остаются лежать на поляне. Так, второму аборигену досталось уже на одну кучу больше — на ту самую, которую принёс первый испытуемый.

### Формат входного файла

В единственной строке через пробел записаны попарно различные целые числа  $x_1$ ,  $x_2$  и  $x_3$  ( $1 \leq x_1, x_2, x_3 \leq 10^{18}$ ) — количество камней в кучах, лежавших на поляне в тот момент, когда профессор Баттеруорт позвал первого аборигена.

### Формат выходного файла

Выведите, скольким аборигенам сможет задать свой глупый вопрос профессор Баттеруорт.

### Пример

<code>input.txt</code>	<code>output.txt</code>
11 5 9	3

Первый абориген покажет на кучи из 11 и 9 камней и принесёт кучу из двух камней. Второй абориген снова укажет на кучи из 11 и 9 камней и принесёт ещё одну кучу из двух камней. Третий абориген укажет на две кучи из двух камней, и эксперименты на этом закончатся.

## Задача Е. Китайский хоккей 2

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Ох уж эти хоккейные фанаты! Они ездят на все матчи любимой команды, не жалея ни времени, ни денег. Вот и болельщики хоккейной команды города Харбина часто собираются небольшой группой в несколько тысяч человек, чтобы поехать со своими кумирами на очередной матч.

Харбинские болельщики хорошо организованы. За каждым из них закреплён уникальный личный номер — целое число от 1 до  $n$ . Перед очередным матчем, на который собрались поехать все  $n$  болельщиков, фанклуб заказал  $m$  автобусов и распределил по ним болельщиков так, чтобы в каждом из автобусов ехало не менее  $a$  и не более  $b$  человек. Чтобы избежать толкучки и неразберихи, все автобусы пронумеровали целыми числами от 1 до  $m$ , а каждому фанату выдали бирку, на которой был написан номер его автобуса. Известно, что в автобусе с бóльшим номером должны были ехать фанаты с бóльшими личными номерами.

Выходец из Вьетнама Ли Си Цын ещё не привык к организованности своих товарищей. Когда он пришёл на место посадки в автобусы, он понял, что забыл дома бирку с номером своего автобуса! Ли Си Цын узнал у нескольких товарищей их личные номера и номера, написанные на их бирках. По этой информации он надеется вычислить номер своего автобуса.

### Формат входного файла

В первой строке через пробел записаны целые числа  $n$ ,  $m$ ,  $a$  и  $b$  ( $2 \leq n \leq 10^5$ ;  $1 \leq a \leq b \leq n$ ;  $2 \leq m \leq n$ ;  $ma \leq n \leq mb$ ). Во второй строке записано целое число  $r$  ( $1 \leq r \leq n$ ) — личный номер Ли Си Цына. В третьей строке записано целое число  $s$  ( $1 \leq s \leq n - 1$ ) — количество фанатов, которых успел опросить Ли Си Цын. В  $i$ -й из следующих  $s$  строк через пробел записаны целые числа  $r_i$  и  $f_i$  ( $1 \leq r_i \leq n$ ;  $1 \leq f_i \leq m$ ) — личный номер болельщика и номер автобуса на его бирке. Все числа  $r_i$  попарно различны, и никакое из них не совпадает с  $r$ .

### Формат выходного файла

Если информация, полученная от опрошенных болельщиков, противоречива, выведите в единственной строке «IMPOSSIBLE». Иначе выведите в первой строке количество вариантов номера автобуса, который мог быть написан на бирке Ли Си Цына, а во второй строке — все эти варианты через пробел в порядке возрастания.

### Пример

input.txt	output.txt
16 4 1 16	2
3	2 3
2	
2 2	
4 3	

## Задача F. Великая команда

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

После того как несколько студентов УрГУ завершили свою спортивную карьеру, в университете возникли трудности с формированием команд. Ветераны спортивного программирования решили вмешаться и создать самую успешную команду за всю историю УрГУ.

Ветераны предположили, что успехи команды в первую очередь зависят от количества друзей членов команды среди АСМ-сообщества УрГУ. После долгих раздумий они разработали *критерий успеха*: из команды никто не должен выделяться, поэтому у всех её членов должно быть одинаковое количество друзей.

Увы, идея ветеранов потерпела крах — выяснилось, что из всех студентов УрГУ, занимающихся спортивным программированием, нельзя выбрать трёх студентов, удовлетворяющих критерию успеха.

Можете ли вы по этой информации восстановить, какие студенты дружат друг с другом?

### Формат входного файла

В первой строке записано единственное целое число  $n$  ( $3 \leq n \leq 200$ ) — количество студентов УрГУ, занимающихся спортивным программированием.

### Формат выходного файла

Если расчёты ветеранов верны, выведите в первой строке целое число  $k$  — количество пар студентов, которые дружат друг с другом, а в следующих  $k$  строках перечислите все эти пары. Студентов следует нумеровать целыми числами от 1 до  $n$ . Если задача имеет несколько решений, выведите любое из них.

Если ветераны ошиблись в расчётах, и задача не имеет решения, в единственной строке выведите «-1».

### Пример

input.txt	output.txt
4	2 1 3 3 4

## Задача G. Код Грея

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Денис, Ваня и Федя собрались на свою первую командную тренировку. Федя рассказал, что выучил алгоритм генерации *кода Грея*:

1. Создадим список из двух элементов:

$$\{0, 1\}.$$

2. Добавим в конец списка все его элементы в обратном порядке:

$$\{0, 1, 1, 0\}.$$

3. К первой половине элементов списка допишем слева 0, ко второй половине элементов списка допишем слева 1:

$$\{00, 01, 11, 10\}.$$

4. Будем повторять шаги 2 и 3 до тех пор, пока длина всех элементов списка не станет равна  $n$ .

Число  $n$  называется *длиной* кода Грея. Так, код длины 3 выглядит следующим образом:  $\{000, 001, 011, 010, 110, 111, 101, 100\}$ .

Когда Денис применил алгоритм Феде, у него получилось, что на  $k$ -й позиции в списке (если нумеровать позиции с нуля) стоит двоичное число  $x$ . Ваня записал на бумажку числа  $k$  и  $x$  в двоичной системе счисления. Спустя много лет эта бумажка попала к вам в руки. К сожалению, некоторые цифры на ней стёрлись за эти годы. Можете ли вы по оставшимся цифрам восстановить числа, которые были на ней записаны?

### Формат входного файла

В первой строке записано число  $k$  в двоичной системе счисления. Стёршиеся цифры обозначены символом «?». Во второй строке в аналогичном формате записано число  $x$ . Длины обоих чисел совпадают и не превосходят  $10^5$ . Числа могут содержать ведущие нули.

### Формат выходного файла

Если по уцелевшим цифрам можно однозначно восстановить числа  $k$  и  $x$ , выведите их, заменив символы «?» на символы «0» и «1». Если существует несколько способов сделать это, выведите «Ambiguity». Если Денис или Ваня ошиблись, и восстановить числа невозможно, выведите «Impossible».

### Примеры

<code>input.txt</code>	<code>output.txt</code>
0?1 0?0	011 010
?00 ??0	Ambiguity
100 100	Impossible

## Задача Н. Чистое программирование

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

В Уральском государственном университете  $n$  компьютерных классов. В субботу, 9 октября, было решено провести в университете сразу  $n$  соревнований по программированию подряд! Организаторы составили расписание — таблицу размером  $n \times n$  из нулей и единиц.  $j$ -е число в  $i$ -й строке равно единице, если  $j$ -й класс задействован в проведении  $i$ -го соревнования, и нулю в противном случае.

В пятницу уборщица Зина напомнила организаторам, что после соревнований ей нужно прибраться в классах. Она сказала, что сразу же после завершения первого соревнования хочет прибраться в первом компьютерном классе, после завершения второго соревнования — во втором классе, и так далее. Конечно, ни во время уборки класса, ни после неё в этом классе уже не могут проходить соревнования.

Председатель жюри согласился с Зиной. За одну операцию он может выбрать пару различных целых чисел  $i$  и  $j$  ( $1 \leq i, j \leq n$ ), поменять в таблице местами  $i$ -ю и  $j$ -ю строку, после чего сразу же поменять местами  $i$ -й и  $j$ -й столбец. До вечера председатель успеет выполнить не более двухсот таких операций. Сможет ли он получить расписание, устраивающее Зину?

### Формат входного файла

В первой строке записано целое число  $n$  ( $2 \leq n \leq 100$ ). Далее следует расписание соревнований:  $n$  строк по  $n$  чисел в каждой.

### Формат выходного файла

Если председатель не успеет исправить расписание, выведите «-1». В противном случае выведите в первой строке количество операций  $t$ , а затем выведите  $t$  строк по два числа в каждой, задающие числа  $i$  и  $j$ , которые должен выбрать председатель для очередной операции.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
3 0 0 0 1 1 0 1 1 0	1 1 3
3 1 1 1 1 1 0 1 0 0	-1



## Задача I. Новое слово Джека

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Джек немногословен. Он не любит повторять одно и то же несколько раз подряд. Поэтому в двоичном слове, которое Джек недавно написал на заборе, нет непустых подстрок вида  $xuxux$ , где  $x$  и  $y$  — двоичные строки (возможно, нулевой длины), а длина  $y$  не превосходит удвоенной длины  $x$ . Например, в слове Джека не могут встречаться подстроки `000` или `1001001`, но могут встречаться подстроки `1010` и `001100110`.

Проходящий мимо лис По спросил Джека, как тот получил своё новое слово. Джек сказал, что сначала на заборе было написано пустое слово, а потом... Дальнейший рассказ Джека содержит только фразы вида:

- «Я приписал 0 (или 1) в начало текущего слова»;
- «Я приписал 0 (или 1) в конец текущего слова»;
- «Я заменил все нули на строку `01`, а все единицы — на строку `10`».

Лису По интересно, но больше ста таких фраз он не вытерпит. Успеет ли Джек завершить свой рассказ?

### Формат входного файла

В единственной строке записано новое слово Джека. Это слово непустое, состоит только из нулей и единиц, а его длина не превосходит  $10^5$ .

### Формат выходного файла

Если для описания своего слова Джеку нужно более ста фраз, выведите «-1». В противном случае выведите любое из возможных описаний. В первой строке выведите количество фраз  $k$  ( $1 \leq k \leq 100$ ). В следующих  $k$  строках перечислите эти фразы в том порядке, в котором их нужно говорить. Если требуется приписать символ  $c$  в начало, выведите «front  $c$ ». Если требуется приписать символ  $c$  в конец, выведите «back  $c$ ». Если требуется заменить 0 на `01`, а 1 на `10`, выведите «double».

### Пример

input.txt	output.txt
011010011	5 back 1 front 0 double double back 1

По рассказу из примера последовательно будут получаться строки: пустая строка, «1», «01», «0110», «01101001», «011010011».

## Задача J. Ядерная гонка

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Западная и Восточная Кукуляндия близки к началу войны. Сверхдержавы борются за превосходство в области ядерного оружия, чтобы добиться доминирования в военной сфере. К сожалению, производство и хранение ядерных боеголовок стоит огромных денег и может легко подорвать бюджеты обеих стран.

Военные аналитики и экономисты Западной Кукуляндии предоставили отчёт, согласно которому страна будет в безопасности, а бюджет — в норме, если к концу  $i$ -го месяца на складах будет храниться ровно  $a_i$  боеголовок. Президент приказал придерживаться этих цифр, и заводы Западной Кукуляндии каждый месяц производят или утилизируют нужное количество боеголовок.

Но разведка Восточной Кукуляндии не дремлет! В начале  $i$ -го месяца агенты Восточной Кукуляндии узнают и передают на родину планы Западной Кукуляндии на ближайшие  $m$  месяцев, то есть, числа  $a_i, a_{i+1}, \dots, a_{i+m-1}$ . Получив эту информацию, диктатор Восточной Кукуляндии сразу же отдаёт приказ изменить текущее количество боеголовок на складах Восточной Кукуляндии на число  $x_i$ . Он выбирает  $x_i$  таким образом, что если Восточная Кукуляндия будет  $m$  месяцев подряд изменять количество боеголовок на  $x_i$ , то в конце каждого месяца у неё будет не меньше боеголовок, чем у Западной Кукуляндии. При этом диктатор тоже заботится о бюджете страны, поэтому выбирает  $x_i$  минимально возможным.

Определите, какие приказы будет отдавать диктатор Восточной Кукуляндии в течение первых  $n$  месяцев. Можете считать, что к началу первого месяца ни Западная, ни Восточная Кукуляндия не располагают ядерным арсеналом.

### Формат входного файла

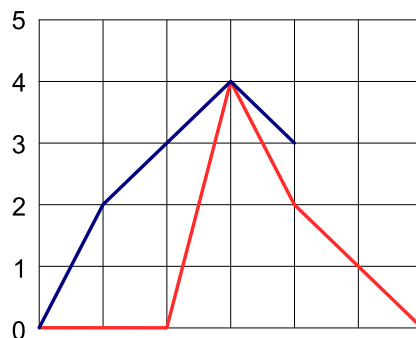
В первой строке через пробел записаны целые числа  $n$  и  $m$  ( $1 \leq n \leq 10\,000$ ;  $1 \leq m \leq 50$ ). Во второй строке через пробел записаны целые числа  $a_1, \dots, a_{n+m-1}$  ( $0 \leq a_i \leq 10^5$ ) — планы Западной Кукуляндии.

### Формат выходного файла

Выведите через пробел целые числа  $x_1, \dots, x_n$ . Число  $x_i$  соответствует приказу, который должен отдать диктатор Восточной Кукуляндии в начале  $i$ -го месяца.

### Пример

input.txt	output.txt
4 3 0 0 4 2 1 0	2 1 1 -1



## Задача К. Шулеры

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

После позорного поражения от Иисуса в боулинг Чувак, Донни и Уолтер решили переквалифицироваться в шулеров. Первым делом они решили научиться свободно обращаться с картами.

Уолтер дал Донни простое задание — разложить колоду из 52 карт на четыре стопки так, чтобы карты в каждой стопке шли по возрастанию достоинства: сверху лежала двойка, сразу под ней — тройка, и так далее. Нижней картой должен был быть туз. Уолтер хотел, чтобы в каждой стопке были карты только одной масти, но забыл про это сказать. В результате масти в стопках Донни оказались перемешаны.

Чувак увидел багровеющее лицо Уолтера и решил исправить ситуацию. Но после пяти «Белых русских», выпитых вчера с горя, он соображал очень туго и мог выполнять только следующие действия:

- взять несколько карт одной масти с вершины стопки и сделать из них новую стопку;
- взять несколько карт одной масти с вершины стопки и положить их на карту, достоинство которой на единицу больше достоинства нижней из взятых карт.

При переносе карт Чувак не менял их порядок.

Помогите Чуваку как можно быстрее выполнить задание Уолтера, пока тот не выхватил «Узи» и не учинил расправу над друзьями.

### Формат входного файла

Входные данные состоят из четырёх строк, описывающих стопки карт. В каждой строке через пробел перечислены 13 карт, составляющих стопку, в порядке сверху вниз (то есть, в порядке возрастания их достоинств). Каждая карта задаётся достоинством и мастью, записанными слитно. Достоинством может быть 2, 3, ..., 9, T (десять), J (валет), Q (дама), K (король), A (туз), мастью — S (пики), C (трефы), D (бубны) или H (червы). Никакая карта не встречается в стопках дважды.

### Формат выходного файла

Выведите минимальное количество действий, которое понадобится выполнить Чуваку, чтобы получить четыре стопки, в каждой из которых лежат карты только одной масти.

### Пример

input.txt												
2C	3C	4C	5C	6C	7C	8C	9C	TC	JC	QC	KC	AC
2S	3S	4S	5S	6S	7S	8D	9D	TD	JD	QD	KD	AD
2D	3D	4D	5D	6D	7D	8S	9S	TS	JS	QS	KS	AS
2H	3H	4H	5H	6H	7H	8H	9H	TH	JH	QH	KH	AH
output.txt												
3												