

## Задача А. Одинокое число

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

*Это интерактивная задача.*

Однажды Маше было нечего делать, и она записала на листе бумаги  $N$  целых чисел:  $a_1, a_2, \dots, a_N$ . Маша недавно изучила алгоритмы сортировки, поэтому она выписала свои числа в неубывающем порядке, то есть  $a_1 \leq a_2 \leq \dots \leq a_N$ .

Также Маша очень любит загадки, поэтому среди ее чисел есть некоторое число  $C$ , которое встречается среди выписанных чисел ровно один раз, а все остальные числа встречаются ровно два раза.

Маша загадала вам загадку — найти «одинокое» число  $C$ . Для этого вы можете не более, чем 42 раза попросить Машу сообщить вам  $i$ -е записанное число.

Маша сообщила вам, что  $1 \leq a_i \leq 10^9$ .

### Протокол взаимодействия

В начале ваше решение должно считать число  $N$  ( $1 \leq N \leq 10^6$ ) — количество записанных Машей чисел.

Затем ваше решение может сделать не более 42 запроса. Для того, чтобы сделать запрос, ваше решение должно вывести его в следующем формате: «?  $i$ » (без кавычек,  $1 \leq i \leq N$ ). Ответом на запрос является число  $a_i$ .

Для того, чтобы вывести ответ, ваше решение должно вывести «!  $C$ », после чего немедленно завершить работу.

Вы должны в точности соблюдать протокол взаимодействия с интерактором, в противном случае решение может получить произвольный вердикт.

При превышении числа запросов вы получите вердикт «Неправильный ответ».

После каждого запроса, в том числе после вывода ответа, вы должны выполнить операцию `flush`.

Для сброса буфера вывода (то есть для операции «`flush`») сразу после вывода запроса и перевода строки нужно сделать:

- `fflush(stdout)` или `cout.flush()` в языке C++;
- `System.out.flush()` в Java;
- `sys.stdout.flush()` в Python;
- `flush(output)` в Pascal;
- смотрите документацию для других языков.

Если вы не сделаете операцию `flush` после какого-либо запроса, ваше решение может получить любой вердикт.

## Пример

стандартный ввод	стандартный вывод
5	? 1
1	? 2
1	? 3
4	? 4
5	? 5
5	! 4

## Замечание

В примере записанные числа равны: 1, 1, 4, 5, 5. Одиноким числом, конечно, является число 4.

## Задача В. Из машины

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

Специальный агент Адам взламывает калькулятор члена секретной масонской организации иллюминатов, чтобы получить доступ к информации о её лидерах. Он уже выяснил, что всего в организацию входит  $n$  человек, каждый из которых имеет свой порядковый номер. Кроме того, у каждого из них есть уровень, представляющий собой целое число от 1 до  $n$  включительно. Уровни всех членов этой организации попарно отличаются. Любой иллюминат может командовать другими иллюминатами более низкого уровня, но вынужден подчиняться иллюминатам более высокого уровня. Адам полагает, что утечка информации об иллюминате наивысшего  $n$ -го уровня может его дискредитировать, поэтому ему интересен иллюминат с уровнем  $n - 1$ , подчиняющийся только иллюминату с наивысшим уровнем.

Доступ к секретной сети иллюминатов получить непросто, поэтому единственный вид запросов, которые может делать Адам — это запросы на сравнение уровней двух иллюминатов с заданными порядковыми номерами. В ответ на такой запрос он получает ответ, больше уровень первого иллюмината, чем уровень второго, меньше, или же их уровни равны. Количество запросов ограничено, но из-за особого отношения калькулятора иллюминатов к степеням двойки Адам может безопасно выполнить целых  $(n + 24)$  запросов, прежде чем его присутствие в сети будет замечено.

### Протокол взаимодействия

Это интерактивная задача. Ваша программа должна общаться с программой жюри, используя для этого стандартные потоки ввода и вывода.

Сначала в отдельной строке записано единственное целое число  $n$  ( $2 \leq n \leq 1000$ ) — количество иллюминатов.

После этого ваша программа может делать запросы на сравнение двух иллюминатов. Для этого в отдельной строке выведите символ «?» и два целых числа через пробел после него — порядковые номера иллюминатов, которых вы хотите сравнить. В ответ на это в отдельной строке будет записан единственный символ «<», если первый иллюминат имеет уровень меньше, чем второй, символ «>», если первый иллюминат имеет уровень больше, чем второй, либо «=», если указанные иллюминаты имеют одинаковый уровень. Вы можете сделать не более чем  $(n + 24)$  таких запросов.

Когда вы узнаете ответ, выведите символ «!» и единственное целое число через пробел — номер иллюмината, имеющего уровень  $n - 1$ .

### Пример

стандартный ввод	стандартный вывод
5	? 1 2
>	? 1 3
>	? 1 4
>	? 1 5
>	? 2 3
>	? 2 4
<	? 4 5
>	! 4

### Замечание

Обратите внимание, что после вывода каждого сообщения ваша программа должна очищать потоковый буфер, чтобы выведенная вами информация дошла до программы жюри: например, это делают вызовы «`fflush(stdout)`» или «`cout.flush()`» в C++, «`System.out.flush()`» в Java, «`Console.Out.Flush()`» в C#, «`flush(output)`» в Pascal, «`sys.stdout.flush()`» в Python.

## Задача С. Хорошо или плохо?

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

Это интерактивная задача!

Просмотрев несколько десятков увлекательных видеороликов про урбанистику и городское планирование, вы внезапно поняли единственный критерий хорошего города: в нем должна быть хорошая межрайонная связность!

Город можно представить как простой неориентированный граф на  $n$  вершинах с  $m$  ребрами. Считается, что в городе плохая межрайонная связность, если хотя бы половина его ребер — мосты, иначе межрайонная связность хорошая.

Вы решили проверить, правда ли, что у вашего города плохая межрайонная связность. Для этого вы можете не более  $3n$  раз задать вопрос вида  $v$ . В ответ на этот запрос вам сообщат какое-то ребро, исходящее из вершины  $v$  (ранее не названное), или  $-1$ , если таких ребер нет.

### Протокол взаимодействия

В первой строке стандартного потока ввода даны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 100\,000$ ) — число вершин и ребер в графе соответственно.

Для того, чтобы узнать очередное ребро, исходящее из  $u$ -й вершины ( $1 \leq u \leq n$ ), нужно вывести «?  $u$ ». После этого ваша программа на вход получит целое число  $v$  ( $-2 \leq v \leq -1$  или  $1 \leq v \leq n$ ) —  $v = a + b - u$ , если существует ребро  $ab$ , которое инцидентно вершине  $u$  и **ещё не было названо**,  $-1$ , если такого ребра не существует и  $-2$ , если вы превысили допустимое число запросов. В последнем случае ваша программа должна немедленно завершиться, в ином случае жюри не гарантирует корректность полученного вами вердикта.

Вам разрешается задать не более  $3n$  вопросов.

Чтобы сообщить, что ответ найден, требуется вывести «! Yes» или «! No», в зависимости от того, плохая или хорошая межрайонная связность в городе («! Yes» — если плохая). В случае положительного ответа выведите  $\lceil \frac{m}{2} \rceil$  строк, по два целых числа  $u_i$  и  $v_i$  в каждой ( $1 \leq u_i, v_i \leq n$ ), обозначающих, что ребро  $(u_i, v_i)$  является мостом. Любое ребро в приведенном списке должно встречаться не более одного раза (кратные ребра считаются различными).

Запрос на вывод ответа не входит в ограничение на  $3n$  запросов.

## Примеры

стандартный ввод	стандартный вывод
3 3	? 3
2	? 1
2	? 2
-1	? 1
3	? 1
-1	? 3
-1	! No
4 4	? 1
2	? 2
3	? 3
2	? 1
-1	? 3
4	? 3
-1	? 2
-1	? 4
-1	! Yes
	1 2
	3 4

## Замечание

Ребро, соединяющее вершины  $u$  и  $v$ , называется мостом, если после его удаления между вершинами  $u$  и  $v$  не существует пути.

## Задача D. Новогодний и прямоугольный

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

Это **интерактивная** задача.

На Новый год Дед Мороз подарил Глебу то, о чём он уже давно мечтал — клетчатый квадрат размером  $n \times n$ . Подарок этот не простой, а с сюрпризом — внутри квадрата Дед Мороз выбрал некоторый непустой прямоугольник, и в каждую клетку этого прямоугольника он положил по мандарину.

Теперь, чтобы получить желанный подарок, Глебу нужно сыграть с Дедом Морозом в очень интересную игру. Глеб должен отгадать, в каком именно прямоугольнике находятся все мандаринки, подаренные Дедом Морозом. Будем считать, что строки и столбцы занумерованы числами от 1 до  $n$  снизу вверх и слева направо. Глеб может производить два типа запросов:

- ?  $x_1 y_1 x_2 y_2$  ( $1 \leq x_1 \leq x_2 \leq n$ ,  $1 \leq y_1 \leq y_2 \leq n$ ) — в ответ на этот запрос Дед Мороз говорит, сколько мандаринок находится в прямоугольнике, левым нижним углом которого является клетка  $(x_1, y_1)$ , а правым верхним — клетка  $(x_2, y_2)$ ;
- !  $x_1 y_1 x_2 y_2$  ( $1 \leq x_1 \leq x_2 \leq n$ ,  $1 \leq y_1 \leq y_2 \leq n$ ) — когда Глеб уверен, что он точно знает, где находятся мандаринки, он должен сделать запрос такого вида, чтобы сообщить свой ответ. При этом  $(x_1, y_1)$  соответствует предполагаемому расположению левого нижнего угла, а  $(x_2, y_2)$  — правого верхнего.

### Формат входных данных

При запуске решения на вход вашей программе подается одно число  $n$  ( $1 \leq n \leq 2 \cdot 10^9$ ) — размер квадрата.

Затем на каждый запрос типа “?” вам будет выдаваться количество мандаринок, находящихся в указанном вами прямоугольнике.

### Формат выходных данных

Вы должны выводить корректные запросы в формате, описанном выше. Последним должен следовать единственный запрос вида “!”, после чего ваша программа должна немедленно завершиться. Ваша программа должна произвести не больше 64 запросов типа “?”. Обратите внимание, что последний запрос, выводящий ответ, не входит в данные 64 запросов.

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер вывода — для этого используйте команды `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`.

### Примеры

стандартный ввод	стандартный вывод
4	
6	? 1 1 4 4
6	? 1 3 4 4
4	? 2 3 4 4
	! 1 3 3 4

### Замечание

Пример в условии иллюстрирует взаимодействие с проверяющей программой. Для прохождения первого теста не обязательно производить такие же запросы, как в примере.

## Задача E. Troll

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

**Эта задача интерактивная!**

Недавно Анакентий ходил на концерт рок-исполнительницы Доры и там познакомился с девушкой по имени Анастасия. Анакентий влюбился по уши и теперь хочет поразить Анастасию своим умением рисовать.

У Анакентия есть огромный лист клетчатой бумаги размером  $10^{18} \times 10^{18}$  клеток. Анакентий хочет на этом листе нарисовать рисунок, представляющий из себя  $n$  закрашенных клеток, причем  $i$ -я из них должна иметь координаты  $(x_i, y_i)$ . При этом Анакентий не собирается дарить весь лист бумаги Насте в качестве рисунка, а хочет вырезать из него часть, где будут содержаться эти  $n$  закрашенных клеток. Формально должны существовать  $(c_1, r_1)$  и  $(c_2, r_2)$  такие, что в подпрямоугольнике с левым верхним углом  $(c_1, r_1)$  и правым нижним  $(c_2, r_2)$  закрашены ровно  $n$  клеток, и их координаты  $(c_1 + x_i, r_1 + y_i)$ .

К сожалению, не все так просто. У Анакентия есть друг Петр, который ни во что не ставит чувства своего Анакентия и поэтому пытается помешать ему нарисовать картину для Насти. После каждых двух закрашенных клеток Анакентием, Петр выбирает одну клетку и портит ее, навсегда делая ее белой.

Вам требуется помочь Анакентию переиграть Петра и все же нарисовать требуемый рисунок.

### Формат входных данных

В первой строке вводится одно число  $1 \leq n \leq 15$  — количество закрашенных клеток в ожидаемом рисунке.

В следующих  $n$  строках вводятся по два числа  $0 \leq x_i, y_i \leq 10^9$  — координаты клеток, оординаты клеток, которые должны стать закрашенными.

### Протокол взаимодействия

Ваша программа может выводить два вида запросов:

- Запрос покраски очередных клеток с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  — `paint x1 y1 x2 y2`
- Запрос о готовности рисунка, верхняя левая клетка которого равна  $(x, y)$  — `end x y`

В ответ вашей программе интерактор выводит сообщение о порче Петром очередной клетки с координатами  $(x, y)$  — `ban x y`

### Пример

стандартный ввод	стандартный вывод
2	
1 2	
2 3	
ban 1 2	paint 1 2 1 4
ban 1 8	paint 1 6 1 8
ban 2 7	paint 2 5 2 7
	end 0 2

## Задача F. Шляпа

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

### Это интерактивная задача.

Имур Ихаков организует клуб по шляпе. На клуб пришло принять участие  $n$  человек, где  $n$  чётно. Имур усадил их всех в круг и провёл жеребьевку, чтобы разбить ребят на пары, но что-то пошло не так. Участники пронумерованы так, что участники  $i$  и  $i + 1$  ( $1 \leq i \leq n - 1$ ) сидят рядом, а также участники  $n$  и  $1$  сидят рядом. Каждому был выдан листочек с числом так, что у ребят, сидящих рядом, эти числа отличаются ровно на единицу. Предполагалось, что игроки с одинаковыми числами образуют пару, но оказалось, что не все числа встречались ровно дважды.

Как известно, удобнее всего объяснять слова партнёру, когда он сидит напротив. Участники с номерами  $i$  и  $i + \frac{n}{2}$  сидят напротив друг друга. Имуре интересно, есть ли люди, сидящие напротив друг друга и имеющие одинаковые числа на своих листочках. Помогите ему найти такую пару людей, если она есть.

Вы можете задавать вопросы вида «какое число написано на листочке у школьника  $i$ ?». Ваша цель — выяснить, есть ли требуемая пара, сделав не более 60 вопросов.

### Формат входных данных

На вход подаётся одно целое чётное число  $n$  ( $2 \leq n \leq 100\,000$ ) — число участников, пришедших на клуб Имуре.

Вам разрешается задать не более 60 вопросов.

### Формат выходных данных

Для того, чтобы узнать число  $i$ -го участника ( $1 \leq i \leq n$ ), нужно вывести «?  $i$ ». После этого ваша программа на вход получит целое число  $a_i$  ( $-10^9 \leq a_i \leq 10^9$ ) — число на листочке  $i$ -го человека.

Чтобы сообщить, что ответ найден, требуется вывести «!  $i$ », где  $i$  — номер любого участника из пары ( $1 \leq i \leq n$ ). Если такой пары участников не существует, выведите «! -1». После этого программа должна завершиться.

Запрос на вывод ответа не входит в ограничение на 60 запросов.

Не забывайте сбрасывать буфер после каждого запроса. Например, на C++ надо использовать функцию `fflush(stdout)`, на Java вызов `System.out.flush()`, на Pascal `flush(output)` и `stdout.flush()` для языка Python.

### Взломы

Используйте следующий формат для взломов:

В первой строке выведите одно чётное целое число  $n$  ( $2 \leq n \leq 100\,000$ ) — количество школьников на клуб Имуре.

Во второй строке выведите  $n$  чисел  $a_i$  ( $-10^9 \leq a_i \leq 10^9$ ) разделённые пробелами, где  $a_i$  — число, которое нужно написать на листочке  $i$ -му школьнику. Любые два соседних элемента, включая  $n$  и  $1$ , должны отличаться на 1 или  $-1$ .

Взламываемое решение не будет иметь прямого доступа к последовательности  $a_i$ .

## Примеры

стандартный ввод	стандартный вывод
8	? 4
2	? 8
2	! 4
6	? 1
1	? 2
2	? 3
3	? 4
2	? 5
1	? 6
0	! -1

## Замечание

Ввод-вывод в примерах демонстрирует пример взаимодействия.

В первом примере были загаданы результаты жеребьёвки соответствующие последовательности 1, 2, 1, 2, 3, 4, 3, 2

Во втором примере была загадана последовательность 1, 2, 3, 2, 1, 0.

## Задача G. Медианная сила

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

Во время проведения эксперимента в космосе вам довелось работать с  $N$  объектами, помеченными числами от 1 до  $N$ . известно, что  $N$  нечетно. Каждый объект обладает определенной, но неизвестной силой, выраженной натуральным числом. Для каждой силы  $Y$  выполняется,  $1 \leq Y \leq N$ . Объект с медианной силой — это такой объект  $X$ , что существует равное количество как объектов, имеющих силу, меньшую, чем  $X$ , так и объектов, имеющих большую силу, чем  $X$ . Вы должны написать программу, которая определяет объект с медианной силой. К сожалению, единственный способ сравнить силы — это использовать устройство, которое по трём различным объектам возвращает объект с медианной силой среди указанных трёх объектов.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $n$  — число объектов в эксперименте.

#### Ограничения:

Для числа  $N$  выполняется, что  $5 \leq N \leq 1499$  и  $N$  нечетно.

- Для номеров объектов  $i$  выполняется  $1 \leq i \leq N$ .
- Для всякой силы  $Y$  выполняется  $1 \leq Y \leq N$ , и все силы различны.
- Разрешено сделать не более чем 7777 запросов определения медианы трёх объектов.

### Формат выходных данных

Чтобы вывести ответ на задачу, выведите его в формате `! ans`, где `ans` — номер объекта с медианной силой.

### Протокол взаимодействия

Чтобы задать запрос описанного в условии формата выведите `? a b c` ( $1 \leq a \neq b \neq c \neq a \leq n$ ), где  $a, b, c$  — номера объектов, про которые вы хотите узнать информацию. Интерагирующая программа возвратит вам одно число —  $a, b$  или  $c$ .

Если возвращаемое значение равно  $-1$ , это означает, что вы превысили максимальное число запросов или сделали некорректный запрос. Ваша программа должна немедленно завершиться (например, вызовом `exit(0)`). Вы получите вердикт «Неправильный ответ», и это будет означать, что вы превысили максимальное число запросов или задали некорректный запрос. Если вы проигнорируете это, то можете получить любой вердикт, так как ваша программа продолжит читать из закрытого потока ввода.

Выше решение получит вердикт «Решение зависло», если вы не будете ничего выводить или забудете сделать операцию `flush` после вывода вопроса или ответа.

Чтобы выполнить операцию `flush`, можете использовать (сразу после вывода запроса и перевода строки):

- `fflush(stdout)` в C++;
- `System.out.flush()` в Java;
- `stdout.flush()` в Python;
- `flush(output)` в Pascal;

Для других языков смотрите документацию.

## Пример

стандартный ввод	стандартный вывод
5	? 1 2 3
3	? 1 2 4
4	? 1 3 4
4	? 4 2 5
4	! 4

## Замечание

Ниже изображено пояснение к примеру из условия:

Номер объекта	1	2	3	4	5
Сила объекта	2	5	4	3	1

## Задача Н. Силовое поле

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Империя обнаружила мятежников на ледяной планете Хот! По сведениям разведки все командование Альянса Повстанцев сейчас скрывается на базе «Эхо», спрятанной в горах на севере этой суровой планеты.

Для того, чтобы окончательно подавить силы восстания, необходимо в ходе стремительной атаки уничтожить эту базу и скрывающихся на ней мятежников. К сожалению, укрытие хорошо укреплено: в частности, его защищает мощное силовое поле, препятствующее бомбардировкам с орбиты. Силовое поле имеет форму выпуклого многоугольника с вершинами в  $N$  специальных станциях-ретрансляторах. Никакие три станции не располагаются на одной прямой.

Перед тем как начинать операцию по уничтожению повстанцев, требуется лишить их базу силового поля, уничтожив эти  $N$  станций точечным бомбометанием. Однако точные координаты этих станций нам неизвестны. Ваша цель — узнать расположение станций-ретрансляторов, чтобы наши войска смогли начать наступление.

На планете введена система координат, устроенная таким образом, что все станции-ретрансляторы находятся в точках с целыми координатами, не превосходящими  $C$  по модулю.

В вашем распоряжении есть зонд-разведчик, оснащенный специальным оборудованием, позволяющим регистрировать станции-ретрансляторы. Если запустить его по прямой над базой повстанцев, по его информации можно будет узнать, сколько станций-ретрансляторов располагаются слева, и сколько — справа от прямой его движения. Станции, находящиеся на его пути, зонд не регистрирует.

С повстанцами надо расправиться как можно скорее: у вас есть время не более чем на  $10^5$  запусков этого зонда. Восстановите по полученной от него информации точные координаты станций-ретрансляторов, чтобы мы могли начать наступление, и Империя вас не забудет!

### Формат входных данных

Это интерактивная задача.

При запуске решения на вход подаются два целых числа  $N$  ( $3 \leq N \leq 1000$ ) и  $C$  ( $5 \leq C \leq 1\,000\,000$ ) — количество станций и ограничение на абсолютную величину их координат.

На каждый запуск зонда-разведчика вводится полученная им информация — два целых числа  $l$  и  $r$ , разделенных пробелом, — количество станций-ретрансляторов слева и справа от траектории его движения соответственно.

### Формат выходных данных

Для запуска зонда выведите строку «?  $x_1 y_1 x_2 y_2$ », где  $(x_1, y_1)$  и  $(x_2, y_2)$  — две точки с целочисленными координатами, лежащие на прямой, по которой должен лететь зонд. Зонд будет лететь в направлении от первой точки ко второй. Точки не должны совпадать. Координаты точек не должны превосходить  $5C$  по модулю.

Как только вы найдете ответ, выведите строку «Ready!», и в следующих  $N$  строках выведите координаты станций в любом порядке. После этого ваша программа должна завершиться.

## Примеры

stdin	stdout
4 5	? -1 3 1 3
0 4	? -1 2 1 2
0 3	? -1 1 0 2
0 3	? -1 0 0 2
0 2	? 0 0 0 2
1 1	? 1 0 1 2
3 1	? 2 0 2 2
3 0	? 3 0 1 2
3 0	Ready!
	0 -1
	2 1
	0 2
	-1 0

## Замечание

В точности соблюдайте формат выходных данных. После вывода каждой строки сбрасывайте буфер 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  $10^5$  запросов запуска зонда. При превышении этого количества, тест будет не пройден с вердиктом «Wrong Answer».

## Задача I. Горилла и ферзи

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

Горилл пришел в клуб настольных игр ЛКШ, но никто не захотел с ним играть. К тому же, все нормальные настолки уже были разобраны.

Горилл нашел квадратную шахматную доску  $n \times n$  и придумал себе занятие. Он хочет расставить на ней  $n$  ферзей так, чтобы никакие два не били друг друга.

Помогите гориллу решить эту непростую задачку.

### Формат входных данных

В единственной строке входного файла находится число  $n$  ( $4 \leq n \leq 200$ ) — размеры доски.

### Формат выходных данных

Выведите  $n$  чисел  $a_i$ :  $a_i$  — это номер горизонтали, на которую горилл поставит ферзя, занимающего  $i$ -ю вертикаль.

Нумерация горизонталей идёт снизу вверх, от 1 до  $n$  (как на обычной шахматной доске).

### Пример

стандартный ввод	стандартный вывод
5	5 3 1 4 2

## Задача J. Горилла и призы

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

Горилл решил раздать  $n$  призов ЛКШатам. Стоимости всех призов различны и выражаются натуральными числами от 1 до  $n$ .

Перед гориллом возникла задача распределить эти призы между  $k$  ЛКШатами так, чтобы все дети получили одинаковое количество призов, и, кроме того, суммарные стоимости призов, полученных разными участниками, совпадали.

Гарантируется, что  $n$  делится на  $k$ .

### Формат входных данных

На вход программы поступают два числа:  $n$  и  $k$  ( $1 \leq n \leq 200$ ,  $1 \leq k \leq 200$ ,  $k$  является делителем  $n$ ).

### Формат выходных данных

Выведите  $k$  строк по  $\frac{n}{k}$  чисел в каждой. В каждой строке должны быть выведены стоимости призов, которые вручаются соответствующему ЛКШатику.

Если распределить призы требуемым образом невозможно, выведите одно число 0.

### Примеры

стандартный ввод	стандартный вывод
8 2	8 1 6 3 2 4 5 7
6 3	3 4 5 2 6 1

## Задача К. Гориллы против ЛКШат

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

В Горилляндии есть  $n$  городов, каждые два из которых соединены дорогой. Эти дороги были построены в давние времена *восточными* и *западными* гориллами. Дороги, которые были построены *восточными* гориллами, вымощены белыми камнями, а те, что построены *западными* — черными. Поскольку гориллы уже изобрели политическую дискриминацию, ни одна *восточная* горилла не может пройти по дороге, вымощенной черными камнями, и ни одна *западная* — по белой дороге.

Когда-то давно ЛКШата решили избрать своих правителей и изгнали горилл из Горилляндии. Однако недавно *восточные* и *западные* гориллы договорились вернуть Горилляндию под свой контроль. Для этого они хотят направить в некоторые города Горилляндии горилл, которые возьмут эти и смежные с ними города под свой контроль.

Точнее, если *восточный* горилл будет направлен в некоторый город, то он возьмет под свой контроль этот город и все города, которые напрямую соединены с ним белыми дорогами. Аналогично, *западный* горилл помимо города, в который он направлен, будет контролировать все города, напрямую соединенные с ним черными дорогами. Для захвата Горилляндии требуется установить контроль над всеми городами.

Однако, при разработке плана захвата обнаружилось две трудности. Во-первых, выяснилось, что горилл согласен принять участие в операции только если все гориллы, которые будут направлены в Горилляндию, будут представлять тот же вид, что и он. То есть либо все участвующие в захвате гориллы должны быть *восточными*, либо все они должны быть *западными*. Во-вторых, общее число горилл, которые могут быть направлены в Горилляндию, не должно превышать  $k$ . Единственная надежда горилл заключается в том, что  $k$  достаточно велико,  $2^k \geq n$ .

Выясните, *восточных* или *западных* горилл следует использовать для захвата Горилляндии, а также в какие города их следует направить.

### Формат входных данных

В первой строке вводятся целые числа  $n$  и  $k$  ( $2 \leq n \leq 256$ ,  $2^k \geq n$ ,  $k \leq n$ ).

Следующие  $n$  строк содержат по  $n$  целых чисел каждая. На  $i$ -й позиции  $i$ -й из этих строк расположено число 0, которое означает, что город не соединен дорогой сам с собой. Для всех  $j \neq i$  число на  $j$ -й позиции  $i$ -й из этих строк равно 1, если  $i$ -й город соединен с  $j$ -м белой дорогой, и равно 2, если они соединены черной дорогой. Числа в строках разделены пробелами.

Гарантируется, что входные данные корректны, то есть если  $i$ -й город соединен с  $j$ -м белой дорогой, то и  $j$ -й соединен с  $i$ -м белой дорогой, аналогично в случае черных дорог.

### Формат выходных данных

Если захватить Горилляндию при заданных условиях невозможно, выведите единственное число 0. В противном случае в первой строке выведите 1, если удастся захватить Горилляндию с использованием восточных горилл, и 2, если требуется использовать западных горилл.

В следующей строке выведите число  $l \leq k$  — количество использованных горилл.

Третья строка должна содержать  $l$  целых чисел — номера городов, в которые следует направить горилл.

Заметьте, что вам не требуется минимизировать  $l$ . Если решений несколько, выведите любое.

## Пример

стандартный ввод	стандартный вывод
8 3	1
0 1 1 1 1 2 2 2	3
1 0 1 1 2 1 2 2	4 6 7
1 1 0 1 2 2 1 2	
1 1 1 0 2 2 2 1	
1 2 2 2 0 2 1 1	
2 1 2 2 2 0 2 1	
2 2 1 2 1 2 0 2	
2 2 2 1 1 1 2 0	

## Задача L. Хорошие раскраски

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

Назовем раскраску клеток таблицы  $n \times m$  хорошей, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет. Иначе говоря, для раскраски не должно быть четверки целых чисел  $x_1, x_2, y_1, y_2$ , что  $1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$ , и клетки  $(x_1, y_1), (x_2, y_1), (x_1, y_2)$  и  $(x_2, y_2)$  покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам  $n, m$  и  $c$  находит любую хорошую раскраску таблицы  $n \times m$  в  $c$  цветов.

### Формат входных данных

В первой строке записаны три целых числа  $n, m, c$  ( $2 \leq n, m \leq 10, 2 \leq c \leq 3$ ).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

### Формат выходных данных

Выведите  $n$  строк по  $m$  чисел в каждой.

В качестве  $j$ -го числа  $i$ -й строки выведите  $a_{i,j}$  — цвет клетки  $(i, j)$  ( $1 \leq a_{i,j} \leq c$ ).

Если есть несколько хороших раскрасок, можно вывести любую из них.

### Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

### Замечание

Примените какой-нибудь неточный алгоритм нахождения минимума функции, например алгоритм отжига.

## Задача М. Кластеризация

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

Дано  $n$  точек на плоскости, обозначающих жилые дома. Необходимо построить  $k$  торговых центров так, чтобы сумма евклидовых расстояний до ближайшего торгового центра от каждого из домов была минимальна. Торговые центры можно располагать в тех же точках, где находятся дома.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 1000$ ) — количество домов и количество торговых центров, которые нужно построить, соответственно.

Следующие  $n$  строк содержат описание жилых домов,  $i$ -я из этих строк содержит два целых числа  $x_i$  и  $y_i$  ( $0 \leq x_i, y_i \leq 10000$ ) — координаты  $i$ -го дома.

### Формат выходных данных

Выведите  $k$  строк, в  $i$ -й из которых находятся координаты  $i$ -го торгового центра. Координаты необходимо выводить ровно с шестью знаками после точки.

### Система оценки

В этой задаче 20 тестов. За каждый тест вы получите  $\lfloor 50 \cdot \min(1, \sqrt[3]{b/c}) \rfloor$ , где  $b$  — авторское суммарное расстояние для этого теста,  $c$  — ваше суммарное расстояние.

### Пример

стандартный ввод	стандартный вывод
7 2	1.750000 3.250000
1 2	5.000000 5.000000
1 4	
2 5	
3 2	
4 4	
5 6	
6 5	