Задача A. Трисочетание

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дан трёхдольный полный взвешенный граф. Все три доли имеют одинаковый размер n. Трёхдольный означает, что нет ребер внутри доли. Полный означает, что между долями проведены все возможные ребра, каждое ребро ровно один раз. Нужно найти трисочетание минимального веса. Трисочетание — n троек индексов (a_i, b_i, c_i) такие, что все a_i различны, все b_i различны, все c_i различны. Вес трисочетания равен $\sum\limits_{i=1..n} (w[0, a_i, b_i] + w[1, b_i, c_i] + w[2, c_i, a_i])$, где $w[i, a, b]$ — вес ребра из a-й вершины i-й доли в b-ю вершину $(i + 1) \bmod 3$ доли.

Формат входного файла

На первой строке число $n \geq 1$.

Следующие n строк содержат матрицу $w[0]$.

Следующие n строк содержат матрицу $w[1]$.

Следующие n строк содержат матрицу $w[2]$.

Гарантируется, что все веса — случайные числа от k до $2k$ для некоторого k. Все веса — целые числа от 0 до 10^5.

Формат выходного файла

На первой строке выведите суммарный вес найденного трисочетания. На каждой из следующих n строк выведите $a_i b_i c_i$ (вершины нумеруются от 0 до $n - 1$). Тройки можно выводить в любом порядке. Если трисочетаний с минимальным суммарным весом несколько, подойдет любое.

Система оценки

Подзадача 1 (25 баллов)	$n \leq 5$.
Подзадача 2 (25 баллов)	$n \leq 9$.
Подзадача 3 (25 баллов)	$n \leq 11$.
Подзадача 4 (25 баллов)	$n \leq 13$.

Примеры

stdin	**stdout**
3 82 100 98 95 71 75 55 97 61 87 99 72 92 59 72 55 80 64 54 57 90 74 60 77 92 50 87	611 2 0 2 1 1 1 0 2 0

Задача B. Множество множеств

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Даны несколько случайных подмножеств множества $A = \{1, 2, \ldots, n\}$. Нужно выбрать минимальное количество подмножеств, которые в объединении дают A. Гарантируется, что объединение всех подмножеств равно A.

Формат входного файла

На первой строке число $n \geq 1$ и число множеств $m \geq 1$. Следующие m строк задают множества. Каждое множество задается числом элементов $k_i \geq 1$ и k_i различными числами от 1 до n. Гарантируется, что множества генерировались следующим образом: фиксируем $c \geq 5$, далее m раз сперва выбираем случайный размер s от 1 до c, далее равновероятно из всех C_n^s вариантов выбираем множество размера s.

Формат выходного файла

На первой строке выведите x — минимальное количество множеств. На следующей строке выведите номера выбранных множеств, x различных чисел от 1 до m. Номера можно выводить в произвольном порядке. Если оптимальных ответов несколько, выведите любой.

Система оценки

Подзадача 1 (25 баллов) $n \leq 20, m \leq 50$.
Подзадача 2 (25 баллов) $n \leq 30, m \leq 100$.
Подзадача 3 (25 баллов) $n \leq 40, m \leq 100$.
Подзадача 4 (25 баллов) $n \leq 50, m \leq 100$.

Примеры

<table>
<tr><th>stdin</th><th>stdout</th></tr>
<tr><td><pre>5 4
4 4 5 1 2
5 3 5 1 2 4
1 2
2 3 5</pre></td><td><pre>1
2</pre></td></tr>
</table>

Задача C. Длинная дорога

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дорога, дорога, осталось немного...

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — найти гамильтонов путь. Гарантируется, что гамильтонов путь в графе есть.

Формат входного файла

На первой строке число вершин $n \geq 2$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Поскольку почти полный граф — совсем не интересный тест, $m \leq 500$.

Формат выходного файла

На первой строке выведите n различных чисел от 1 до n — вершины гамильтонового пути в порядке прохода по ним. Начинать и заканчивать можно в любой вершине. Если гамильтоновых путей несколько, выведите любой.

Система оценки

Подзадача 1 (20 баллов) $n \leq 26$.
Подзадача 2 (20 баллов) $n \leq 35$.
Подзадача 3 (20 баллов) $n \leq 50$.
Подзадача 3 (20 баллов) $n \leq 70$.
Подзадача 4 (20 баллов) $n \leq 100$.

Примеры

stdin	stdout
5 8 3 1 2 5 5 4 3 4 1 4 3 5 3 2 1 2	1 4 3 5 2

Задача E. Корни

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дано целое число $n \geq 1$. Нужно найти такое g, что для любого a: $gcd(a, n) = 1, 1 \leq a < n \quad \exists x \colon g^x = a \mod n$. Напомним, что $gcd(a, b)$ — наибольший общий делитель чисел a и b.

Формат входного файла

Внимание, мультитест!

На каждой строке число n ($2 \leq n \leq 10^{12}$).

Сколько тестов, мы вам не скажем, но все в рамках приличия.

Формат выходного файла

Для каждого n на отдельной строке выведите g ($1 \leq g < n$) или -1, если такого g не существует.

Система оценки

Подзадача 1 (15 баллов) $n \leq 10^3$.
Подзадача 2 (15 баллов) $n \leq 10^5$.
Подзадача 3 (35 баллов) $n \leq 10^9$.
Подзадача 4 (35 баллов) $n \leq 10^{12}$.

Примеры

stdin	stdout
5 10 9 15	2 3 2 -1

Задача F. Умножение многочленов

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Даны многочлены $P(x)$ и $Q(x)$, найдите $P(x)Q(x)$. $Q(x) = x^k - 1$.
У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m - 1$.
Все вычисления происходят по модулю m, m — простое от 2 до $10^9 + 7$.

Формат входного файла

Первая строка содержит целые числа n, k, m.
Следующая строка содержит n чисел $p_0, p_1, \ldots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Формат выходного файла

Выведите $n + k$ целых чисел от 0 до $m - 1$: $b_0, b_1, \ldots, b_{n+k-1}$,
такие, что $P(x)Q(x) = \sum b_i x^i \mod m$.

Система оценки

Подзадача 1 (50 баллов) $1 \le n, k \le 10^3$.
Подзадача 2 (50 баллов) $1 \le n, k \le 10^5$.

Примеры

stdin	**stdout**
`4 1 7` `1 2 3 4`	`6 6 6 6 4`

Задача G. Деление многочленов

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Даны многочлены $P(x)$ и $Q(x)$, найдите такие $A(x)$ и $R(x)$, что $P(x) = A(x)Q(x) + R(x)$ и $degR < degQ$. $Q(x) = x^k - 1$. У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m-1$. Все вычисления происходят по модулю m, m — простое от 2 до $10^9 + 7$.

Формат входного файла

Первая строка содержит целые числа n, k, m.

Следующая строка содержит n чисел $p_0, p_1, \ldots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Формат выходного файла

На первой строке выведите $t = \max(n-k, 1)$ целых чисел от 0 до $m-1$: $a_0, a_1, \ldots, a_{t-1}$. На второй строке k целых чисел от 0 до $m-1$: $r_0, r_1, \ldots, r_{k-1}$. Все эти числа должны обладать свойством, что $P(x) = Q(x)(\sum a_i x^i) + \sum r_i x^i \mod m$.

Система оценки

Подзадача 1 (50 баллов) $1 \le n, k \le 10^3$.
Подзадача 2 (50 баллов) $1 \le n, k \le 10^5$.

Примеры

stdin	stdout
3 1 7 1 5 1	6 1 0
3 1 7 2 4 1	5 1 0
3 1 7 0 4 1	5 1 5

Задача H. Ребра добавляются, граф растет

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

В неориентированный граф последовательно добавляются новые ребра. Изначально граф пустой. После каждого добавления нужно говорить, является ли текущий граф двудольным.

Формат входного файла

На первой строке n — количество вершин, m — количество операций "добавить ребро". Следующие m строк содержат пары чисел от 1 до n — описание добавляемых ребер.

Формат выходного файла

Выведите в строчку m нулей и единиц. i-й символ должен быть равен единице, если граф, состоящий из первых i ребер, является двудольным.

Система оценки

Подзадача 1 (25 баллов)	$1 \le n, m \le 1\,000$.
Подзадача 2 (50 баллов)	$1 \le n, m \le 50\,000$.
Подзадача 3 (25 баллов)	$1 \le n, m \le 300\,000$.

Примеры

stdin	**stdout**
3 3 1 2 2 3 3 1	110

Задача I. Сумма всего подряд

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	128 Мб

Дан случайный неориентированный граф. Нужно для каждого множества вершин A посчитать $f(A)$, количество независимых подмножеств вершин B: $B \subseteq A$. Множество вершин B называется независимым, если в графе нет ребра, оба конца которого лежат в множестве B.

Формат входного файла

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходного файла

Каждому множеству A можно сопоставить целое число $b(A)$, двоичная запись которого соответствует наличию элементов в множестве A.

Пример: $n = 5, A = \{1, 2, 5\}, b(A) = 2^0 + 2^1 + 2^4 = 19$.

Выведите $\sum\limits_{A} f(A) 2^{b(A)} \mod (10^9 + 7)$

Система оценки

Подзадача 1 (25 баллов) $1 \leq n \leq 10$.
Подзадача 2 (25 баллов) $1 \leq n \leq 16$.
Подзадача 3 (30 баллов) $1 \leq n \leq 20$.
Подзадача 4 (20 баллов) $1 \leq n \leq 23$.

Примеры

stdin	stdout
3 1 1 2	1221

Пояснение

Независимыми являются множества вершин $\{\}, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}$.

$A = \{\}$	$f(A) = 1$	$b(A) = 0$
$A = \{1\}$	$f(A) = 2$	$b(A) = 2^0 = 1$
$A = \{2\}$	$f(A) = 2$	$b(A) = 2^1 = 2$
$A = \{1, 2\}$	$f(A) = 3$	$b(A) = 2^0 + 2^1 = 3$
$A = \{3\}$	$f(A) = 2$	$b(A) = 2^2 = 4$
$A = \{1, 3\}$	$f(A) = 4$	$b(A) = 2^0 + 2^2 = 5$
$A = \{2, 3\}$	$f(A) = 4$	$b(A) = 2^1 + 2^2 = 6$
$A = \{1, 2, 3\}$	$f(A) = 6$	$b(A) = 2^0 + 2^1 + 2^2 = 7$

$1 \cdot 2^0 + 2 \cdot 2^1 + 2 \cdot 2^2 + 3 \cdot 2^3 + 2 \cdot 2^4 + 4 \cdot 2^5 + 4 \cdot 2^6 + 6 \cdot 2^7 = 1221$

Задача J. Все клики

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Макс, опытнейший игрок в доту, все кликал и кликал.

Дан случайный неориентированный граф G из n вершин и m ребер. Подкликой называется такое подмножество вершин A: $\forall a, b \in A, a \neq b \quad \exists$ ребро (a, b). Подклика A называется максимальной по включению, если $\nexists$ подклики $B \supset A$: $|B| > |A|$. $K(G)$ — количество максимальных по включению подклик в графе G. Ваша задача — посчитать $min(K(G), 10^6)$.

Формат входного файла

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.
Следующие m строк содержат пары чисел от 1 до n — ребра графа.
В графе нет ни петель, ни кратных ребер.

Формат выходного файла

Единственное число — $min(K(G), 10^6)$.

Система оценки

Подзадача 1 (15 баллов)	$n = 20$.
Подзадача 2 (15 баллов)	$n = 25$.
Подзадача 3 (20 баллов)	$n = 30$.
Подзадача 4 (15 баллов)	$n = 40$.
Подзадача 5 (20 баллов)	$n = 50$.
Подзадача 6 (15 баллов)	$n = 60$.

Примеры

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	2

Задача K. Макс клика

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Макс, опытнейший игрок в доту, постоянно кликал.

Дан случайный неориентированный граф G из n вершин и m ребер.
Подкликой называется такое подмножествао вершин A: $\forall a, b \in A, a \neq b \quad \exists$ ребро (a, b).
Ваша задача — найти подклику A: $|A|$ максимально.

Формат входного файла

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.
Следующие m строк содержат пары чисел от 1 до n — ребра графа.
В графе нет ни петель, ни кратных ребер.

Формат выходного файла

На первой строке выведите k — количество вершин в максимальной подклике. На следующей строке k целых чисел от 1 до n — номера вершин в подклике. Вершины можно выводить в любом порядке. Если максимальных подклик несколько, выведите любую.

Система оценки

Подзадача 1 (10 баллов) $n \leq 20$.
Подзадача 2 (10 баллов) $n \leq 25$.
Подзадача 3 (20 баллов) $n \leq 40$.
Подзадача 4 (20 баллов) $n \leq 60$.
Подзадача 5 (20 баллов) $n \leq 70$.
Подзадача 6 (20 баллов) $n \leq 80$.

Примеры

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 1 3 4 5

Задача L. Учимся красить

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — покрасить его вершины в минимальное количество цветов таким образом, чтобы смежные вершины были покрашены в разные цвета.

Формат входного файла

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходного файла

На первой строке выведите k — минимальное количество цветов. На следующей строке n целых чисел от 1 до k — цвета вершин. Если оптимальных раскрасок несколько, выведите любую.

Система оценки

Подзадача 1 (10 баллов) $n \leq 15$.
Подзадача 2 (15 баллов) $n \leq 20$.
Подзадача 3 (15 баллов) $n \leq 30$.
Подзадача 4 (15 баллов) $n \leq 40$.
Подзадача 5 (15 баллов) $n \leq 50$.
Подзадача 6 (15 баллов) $n \leq 60$.
Подзадача 7 (15 баллов) $n \leq 70$.

Примеры

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 1 1 2 3 4

Задача M. Проще всего красить в три цвета

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — покрасить его вершины в три цвета таким образом, чтобы смежные вершины были покрашены в разные цвета. Гарантируется, что покрасить граф в три цвета возможно.

Формат входного файла

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходного файла

Вывести n целых чисел от 1 до 3 — цвета вершин. Если требуемых раскрасок несколько, выведите любую.

Система оценки

Подзадача 1 (10 баллов)	$n \leq 20$.
Подзадача 2 (15 баллов)	$n \leq 30$.
Подзадача 3 (10 баллов)	$n \leq 45$.
Подзадача 4 (15 баллов)	$n \leq 64$.
Подзадача 5 (15 баллов)	$n \leq 90$.
Подзадача 6 (10 баллов)	$n \leq 120$.
Подзадача 7 (10 баллов)	$n \leq 170$.
Подзадача 8 (15 баллов)	$n \leq 250$.

Примеры

stdin	**stdout**
5 5 1 2 2 3 3 1 4 5 1 5	1 2 3 2 3

Задача N. SAT USAT

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Широко известна задача **3-SAT**. Решите её. Гарантируется, что решение существует. Формулировка **3-SAT**: даны m клозов, состоящих из трех условий вида $x_i = 0$ и $x_i = 1$, нужно подобрать значения n булевых переменных таким образом, чтобы в каждом клозе хотя бы одно из трёх условий было верно.

Формат входного файла

На первой строке число переменных n и клозов m ($1 \le m \le \min(n^2, 1000)$).

Каждая из следующих m строк содержит числа i, e_1, j, e_2, k, e_3 и задает условия $x_i = e_1 \vee x_j = e_2 \vee x_k = e_3$. $0 \le e_1, e_2, e_3 \le 1; 1 \le i, j, k \le n$. **Все клозы случайны, тем не менее гарантируется, что решение существует**.

Формат выходного файла

Выведите строку из n нулей и единиц — значения переменных. Если у данной задачи **3-SAT** есть несколько решений, выведите любое.

Система оценки

Подзадача 1 (10 баллов)	$n \le 20$.
Подзадача 2 (15 баллов)	$n \le 30$.
Подзадача 3 (15 баллов)	$n \le 40$.
Подзадача 4 (15 баллов)	$n \le 50$.
Подзадача 5 (15 баллов)	$n \le 70$.
Подзадача 6 (15 баллов)	$n \le 100$.
Подзадача 7 (15 баллов)	$n \le 160$.

Примеры

stdin	stdout
2 3 1 0 1 0 1 0 2 0 2 1 1 1 1 1 2 1 1 1	01

Пояснение

$(\underline{x_1 = 0} \vee \underline{x_1 = 0} \vee \underline{x_1 = 0}) \wedge (x_2 = 0 \vee \underline{x_2 = 1} \vee x_1 = 1) \wedge (x_1 = 1 \vee \underline{x_2 = 1} \vee x_1 = 1)$

В каждом клозе подчеркнуты истинные условия.

Задача O. Почти двудольный зверь

Вход:	`stdin`
Выход:	`stdout`
Ограничение по времени:	
Ограничение по памяти:	512 Мб

Дан неориентированный, неслучайный граф. Вершины графа разбиты на три множества A, B, X. $|X| = k \le 10$. Между вершинами из A нет ребер, между вершинами из B также нет ребер. Ваша задача выделить минимальное по размеру множество вершин Y, что после выкидывания этого множества граф становится двудольным. Заметьте, что $|Y| \le k$, так как при выкидывании X граф точно станет двудольным.

Формат входного файла

На первой строке число вершин $n \ge 1$, число ребер $m \ge 1$ и $k \ge 1$ — размер множества X, $klen$. Следующие m строк содержат пары чисел от 1 до n — ребра графа. В графе нет ни петель, ни кратных ребер. Вершины из A и B нумеруются числами от 1 до $n - k$. Вершины множества X нумеруются от $n - k + 1$ до n. Номера вершин A и B перемешаны. Специально, чтобы Вам было... удобно.

Формат выходного файла

В первой строке выведите l — размер множества Y, на второй строке l различных целых чисел от 1 до n (элементы множества Y). Если оптимальных множеств несколько, выведите любое.

Система оценки

Подзадача 1 (20 баллов) $n \le 20, k \le 10, m \le 90$.
Подзадача 2 (40 баллов) $n \le 100, k \le 10, m \le 200$.
Подзадача 3 (40 баллов) $n \le 1000, k \le 10, m \le 2000$.

Примеры

stdin	stdout

`7 10 2`
`1 6`
`1 3`
`1 7`
`2 3`
`2 7`
`3 4`
`3 6`
`4 5`
`4 7`
`5 6`
	`1`
	`1`

`9 15 2`
`1 2`
`1 3`
`1 5`
`1 7`
`2 4`
`3 4`
`3 6`
`4 8`
`4 9`
`5 9`
`5 8`
`6 8`
`6 9`
`7 8`
`7 9`
	`1`
	`1`