

Задача А. Леденящая игра

Имя входного файла: `game.in`
Имя выходного файла: `game.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Чтобы попасть в команду к Шкиперу пингвин должен пройти ряд испытаний: полоса препятствий от Шкипера, спарринг с Рико, расшифровка кода от Прапора и задача от Ковальски.

Вы, пингвин-новобранец, успешно дошли до последнего испытания. Ковальски предлагает вам сыграть в следующую игру. Вам дается m наборов разноцветных льдинок, каждая одного из n цветов. Различные цвета обозначаются различными прописными буквами латинского алфавита. Вы можете взять какое-то подмножество этих наборов при условии, что льдинка каждого цвета будет встречаться не более одного раза в этом подмножестве. Пусть вы выбрали k наборов с индексами i_1, i_2, \dots, i_k , тогда ваш выигрыш составляет $\sum_{j=1}^k l_{i_j} - k$ баллов, где l_{i_j} — количество льдинок в наборе i_j .

Ковальски требует найти подмножество с максимальным количеством баллов.

От вас требуется найти любое подмножество, подходящее под условия Ковальски.

Формат входных данных

В первой строке входного файла находится число n ($1 \leq n \leq 17$) — количество различных цветов. Вторая строка входного файла содержит число m ($1 \leq m \leq 200000$) — количество различных наборов льдинок. В следующих m строках перечислены сами наборы. Набор с номером i задаётся строкой из первых n строчных латинских букв. Длина каждой строки не больше 17 символов.

Формат выходных данных

В первой строке выходного файла выведите k — количество наборов в ответе. Во второй строке выходного файла выведите k чисел — индексы наборов, входящих в ответ, в произвольном порядке.

Примеры

<code>game.in</code>	<code>game.out</code>
1 3 aaa aaaa a	0
1 2 aaa aaaa	0
3 3 aba ab c	1 2

Задача В. Раскраска графа

Имя входного файла: coloring.in
Имя выходного файла: coloring.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 32 мегабайта

Дан граф из n вершин, раскрасьте его в минимально возможное число цветов так, чтобы никакие две вершины, соединенные ребром, не были одного цвета.

Формат входных данных

В первой строке содержится число t — количество тестовых примеров ($1 \leq t \leq 5$).

Далее содержится t тестовых случаев, заданных в следующем формате:

В первой строке записаны числа n и m — количество вершин и ребер соответственно ($1 \leq n \leq 17$, $0 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Затем идет m строк, в которых содержится по два числа $v_i u_i$, что означает, что вершины v_i и u_i соединены ребром ($1 \leq v_i, u_i \leq n, v_i \neq u_i$).

Гарантируется, что все ребра в каждом тестовом случае различны.

Формат выходных данных

Для каждого тестового случая в первой строке выведите минимальное число цветов k .

Во второй строке выведите n чисел a_i — цвета вершин ($1 \leq a_i \leq k$).

Примеры

coloring.in	coloring.out
3	3
3 3	3 2 1
1 2	2
2 3	1 2 2 1 1
3 1	3
5 3	1 3 1 1 2 1
2 1	
3 1	
4 2	
6 7	
1 2	
1 5	
2 5	
2 3	
2 4	
5 6	
5 4	

Задача С. Медианная сумма

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

Вам дан массив из N чисел A_1, A_2, \dots, A_N .

Рассмотрим суммы всех непустых подмножеств этого массива. Это $2^N - 1$ сумм, нечетное число.

Пусть $S_1, S_2, \dots, S_{2^N-1}$ это список всех этих сумм в неубывающем порядке. Найдите медиану этого массива, то есть число $S_{2^{N-1}}$.

Формат входных данных

В первой строке входных данных находится число N ($1 \leq N \leq 2000$). В следующей строке находится элементы массива A_i ($1 \leq A_i \leq 2000$).

Формат выходных данных

Выведите единственное число — медиану отсортированного массива сумм.

Примеры

стандартный ввод	стандартный вывод
3 1 2 1	2
1 58	58

Задача D. Таблицы Юнга

Имя входного файла: `young.in`
Имя выходного файла: `young.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Нужно найти число способов расставить числа от 1 до N внутри диаграммы Юнга площади N так, чтобы числа внутри каждой строки и каждого столбца возрастали. Каждое число от 1 до N нужно использовать ровно один раз.

$$1 \leq N \leq 50$$

Формат входных данных

Число строк диаграммы Юнга k . Далее k длин строк: $a_1 \geq a_2 \geq \dots \geq a_k$.

Число 1 должно оказаться в первой клетке первой строки.

Первый столбец диаграммы имеет высоту k . Высоты столбцов, также как и длины строк, убывают.

Формат выходных данных

Одно целое число — количество способов расставить числа от 1 до N по модулю $10^9 + 7$, где

$$N = \sum_{i=1}^k a_i.$$

Примеры

<code>young.in</code>	<code>young.out</code>
2 2 2	2
4 4 3 2 1	768

Задача Е. Интересное число

Имя входного файла: `number.in`
Имя выходного файла: `number.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Для заданного числа n найдите наименьшее положительное целое число с суммой цифр n , которое делится на n .

Формат входных данных

Во входном файле содержится целое число n ($1 \leq n \leq 1000$).

Формат выходных данных

Выходной файл должен содержать искомое число. Ведущие нули выводить не разрешается.

Примеры

<code>number.in</code>	<code>number.out</code>
1	1
10	190

Задача F. Разбиение на пути

Имя входного файла: `vertex-partition.in`
Имя выходного файла: `vertex-partition.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан ориентированный граф. Найти число способов разбить все его вершины на простые пути. Каждая вершина должна лежать ровно в одном пути, каждый путь содержит не менее двух вершин. Разбиения на пути различны, если различны множества использованных рёбер.

Формат входных данных

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 17$, $M \leq n(n - 1)$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел a, b — номерами начальной и конечной вершин соответственно. Рёбра не повторяются, у каждого ребра $a \neq b$.

Формат выходных данных

Выведите число разбиений вершина графа на пути.

Примеры

<code>vertex-partition.in</code>	<code>vertex-partition.out</code>
4 3 1 2 2 3 3 4	2
4 6 1 2 2 3 3 4 4 1 3 1 2 4	8