

Problem A. 1024 Stack Edition

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 512 мегабайт

Недавно девочка Ксюша купила в онлайн-магазине приложение «1024 Stack Edition» — пошаговую аркаду. Как можно догадаться из названия, игра разворачивается вокруг некоторого стека, наполненного степенями двойки. По правилам игры на каждом ходу игрок может выполнить одно из трёх действий над стеком:

- Добавить случайное число на вершину стека.
- Убрать число с вершины стека.
- Если два числа на вершине стека равны, то заменить их на их сумму.

Если игрок добавляет случайное число на вершину стека, то с вероятностью p это число будет единицей, а с оставшейся вероятностью $(1 - p)$ — двойкой, независимо от других сгенерированных случайных чисел. Если игрок выбирает убрать число с вершины стека, он платит штраф в одну монету. Обратите внимание, что все остальные операции, кроме удаления числа с вершины стека, не приводят к уплате штрафа. Для того чтобы выиграть, игроку необходимо привести стек в такое состояние, что в нём находится лишь один элемент — число 1024. При этом чем меньше штрафа он заплатит — тем лучше.

Как-то раз Ксюша оставила свой телефон без присмотра, а по возвращении обнаружила, что кто-то играл в «1024 Stack Edition». Ксюше не интересно, кто это был и каковы были его мотивы. Ей интересно узнать, каково математическое ожидание штрафа, который лично ей придётся заплатить при оптимальной игре, если она доиграет партию. Ваша задача, как вы могли уже догадаться, посчитать это число.

Input

В первой строке ввода содержатся два целых числа — исходный размер стека N ($0 \leq N \leq 100\,000$) и вероятность p ($0 < p \leq 100$), указанная в **процентах**. Во второй строке через пробел указаны N целых чисел s_i ($s_i = 2^h$, где h целое и $0 \leq h \leq 10$), которые находятся в стеке, начиная с самых «глубоких» элементов (вершина стека — это последнее число строки). Обратите внимание, что стек может быть пустым, тогда во второй строке не содержится ни одного числа. Также заметьте, что размер стека во время игры ничем не ограничен, задан лишь начальный его размер.

Output

В единственной строке следует вывести искомое число — математическое ожидание количества монет, которое придётся заплатить Ксюше, доигрывая заданную во вводе партию. Абсолютная или относительная погрешность ответа не должна превосходить 10^{-6} .

Examples

stdin	stdout
10 50 512 256 128 64 32 16 8 4 2 1	1.000000000
9 50 512 256 128 64 32 16 8 4 2	0.500000000

Problem B. Строчная задача

Input file: `stdin`
Output file: `stdout`
Time limit: 4 секунды
Memory limit: 512 мегабайт

Поисковые запросы — что может быть лучше? Поле для ввода вопроса, кнопка «найти» — и весь интернет выстраивается в упорядоченный список ответов. Миллионы поисковых запросов сохраняются каждый день, чтобы в будущем можно было отвечать на них быстрее и лучше, но этого мало! Важно помочь пользователю быстрее вводить сам поисковый запрос. Например, можно подсказывать популярные или интересные пользователю варианты запросов, упорядочивая их в соответствии с предпочтениями. Но что делать, если мы ничего не знаем о том, кто задаёт вопрос?

Вы работаете над алгоритмом, который будет предлагать две подсказки в лексикографическом порядке. Ну, почти в лексикографическом.

Известно, что пользователи могут допустить опечатки при вводе запроса. Скажем, что строка *A* *почти меньше* строки *B*, если строка *A* лексикографически меньше строки *B*, или же если в строке *A* можно заменить одну букву на какую-то другую маленькую букву английского алфавита так, чтобы полученная строка оказалась лексикографически меньше строки *B*.

Вам даны *N* слов-подсказок. Посчитайте количество пар слов, первое из которых почти меньше второго. Количество пар нужно вычислять с учётом порядка: если два слова являются взаимно *почти меньшими*, то учитывайте обе пары.

Input

В первой строке содержится целое число *N* ($1 \leq N \leq 10^6$) — количество слов-подсказок. Следующие *N* строк содержат слова, состоящие только из маленьких букв английского алфавита. Суммарная длина всех слов не превосходит 10^6 .

Output

Выведите искомое количество пар.

Examples

stdin	stdout
3 a b c	4
2 aab aba	2
2 aa aaa	1

Problem C. Кубик

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 512 мегабайт

Дан куб, граням которого поставлены в соответствие различные целые числа от 1 до 6.

Также дано гексамино — фигура из шести клеток на клетчатой плоскости, связанная по стороне. Клеткам гексамино также поставлены в соответствие различные целые числа от 1 до 6.

Определите, можно ли из заданного гексамино сделать куб, который можно будет повернуть так, чтобы нумерация его граней совпала с нумерацией граней заданного куба. Разрешается сгибать гексамино по границам клеток, но не разрезать его.

Input

В первой строке ввода заданы шесть попарно различных целых чисел от 1 до 6 — нумерация граней куба, перечисленных в следующем порядке: передняя, задняя, левая, правая, верхняя, нижняя.

В последующих строках задаётся гексамино в виде массива из $K \times N$ клеток: каждая из последующих K строк содержит ровно N цифр от 0 до 6 без пробелов. Нулям соответствуют пустые клетки, ненулевым цифрам — соответствующие им клетки гексамино.

Гарантируется, что образованная ненулевыми цифрами фигура связна по стороне, что каждая из ненулевых цифр от 1 до 6 встречается ровно один раз, и что любая из K строк и любой из N столбцов массива содержит хотя бы одну ненулевую цифру.

Output

Выведите «Yes», если сложить куб, нумерация граней которого совпадает с заданной, возможно, и «No» в противном случае.

Examples

stdin	stdout
1 2 3 4 5 6 0100 3546 0200	Yes
1 6 2 5 3 4 1020 3546	No

Problem D. НЛО

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 512 мегабайт

Ночью в озеро, расположенное неподалёку от Васиного дома, упал НЛО.

НЛО представлял собой каркас N -мерного прямоугольного параллелепипеда, собранный из титановых рёбер, причём все рёбра имели целые длины. В момент крушения соединения между рёбрами разрушились, в отличие от самих рёбер, которые оказались на дне и на берегу озера.

С утра Вася пришёл на берег озера и нашёл K титановых стержней. Предполагая, что эти стержни являются рёбрами каркаса НЛО и что, возможно, часть рёбер ещё находится под водой, определите наименьшую возможную размерность пространства N , из которого прибыл НЛО.

Input

Первая строка ввода содержит целое число K ($1 \leq K \leq 10^6$) — количество найденных Васей стержней. Вторая строка содержит K целых чисел a_i ($1 \leq a_i \leq 10^6$) — длины стержней, разделённые пробелами.

Output

Выведите одно целое число — наименьшую возможную размерность пространства N , из которого прибыл НЛО.

Examples

stdin	stdout
3 1 2 3	3
3 1 2 2	2

Problem E. Недополняемое паросочетание

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 512 мегабайт

Артёмка очень любит паросочетания.

Паросочетание в неориентированном графе — это набор попарно несмежных рёбер.

Паросочетание называется *недополняемым*, если невозможно добавить в паросочетание ещё одно ребро, не удаляя при этом ни одно из уже включённых рёбер.

Неориентированный граф называется *двудольным*, если его вершины можно разбить на два множества так, чтобы любое ребро графа соединяло вершины из разных множеств.

Артёмке дан двудольный граф. Его задача — найти количество недополняемых паросочетаний в нем по модулю $1\,000\,000\,007$ ($10^9 + 7$). Ваша задача — помочь ему.

Input

В первой строке записано три целых числа n_1 , n_2 и m ($1 \leq n_1 \leq 10$, $1 \leq n_2 \leq 100$, $0 \leq m \leq n_1 \cdot n_2$) — количества вершин в двух долях графа и количество рёбер, соответственно. В следующих m строках содержится описание рёбер графа. В каждой строке записано по два целых числа v_1 и v_2 ($1 \leq v_1 \leq n_1$, $1 \leq v_2 \leq n_2$), означающих, что очередное ребро соединяет вершину v_1 первой доли и вершину v_2 второй доли. Гарантируется, что граф не содержит кратных рёбер.

Output

Выведите одно число — количество недополняемых паросочетаний по модулю $1\,000\,000\,007$ ($10^9 + 7$).

Examples

stdin	stdout
2 3 4 1 1 1 2 1 3 2 2	3
2 2 4 1 1 1 2 2 1 2 2	2

Problem F. Музыкальный мир

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 512 мегабайт

В обновлённом сервисе Яндекс.Музыки музыкальные рекомендации стали ещё лучше, чем раньше! Они учитывают каждый прослушанный пользователем трек с самого начала использования сервиса, и в зависимости от уже прослушанных — увеличивают пул рекомендаций. Плеер включает пользователю случайный из рекомендованных треков, и так целый день.

Но сервера Яндекса такие быстрые, что могут строить рекомендации не только для того трека, которую сейчас слушает пользователь, но и для всех вариантов случайно выбранных следующих треков.

Пользователь выбирает первый трек, а дальше алгоритм автоматически строит рекомендации на N песен вперёд. Из каждой песни i -го поколения может получиться не более K_i ($1 \leq K_i \leq 5$) новых песен в $(i + 1)$ -м поколении, причём известна вероятность $p_{i,j}$ того, что на основании песни i -го поколения будет предсказано j новых песен для каждого j от 0 до K_i . Предсказания на основании каждой отдельной песни строятся независимо от других песен того же поколения.

Так как экспериментальное исследование производилось на жанре инди, все рекомендованные треки различны. Для того, чтобы измерить разнообразие сбора статистики о похожести песен, похожесть измеряется для каждой пары треков. Чтобы помочь разработчикам Яндекса оценить затраты по памяти для этого измерения, вычислите математическое ожидание количества пар треков в пуле рекомендаций в $(N + 1)$ -м поколении.

Input

В первой строке задано число поколений N ($1 \leq N \leq 10^5$), на которые производится предподсчёт.

В следующих N строках задано распределение вероятностей числа предсказанных дорожек для каждого поколения. В i -й из этих строк задано число K_i ($1 \leq K_i \leq 5$) — максимальное число возможных предсказаний для i -го поколения. После него дано $(K_i + 1)$ целое число $a_{i,j}$ ($1 \leq a_{i,j} \leq 1000$) — ненормированная вероятность того, что у песни i -го поколения будет j предсказаний для $(i + 1)$ -го поколения. По ним вероятности $p_{i,j}$ вычисляются как
$$\frac{a_{i,j}}{\sum_{t=0}^{K_i} a_{i,t}}.$$

Первым поколением является первая песня, вторым — песни, рекомендованные из-за первой песни, третьим — песни, рекомендованные из-за каждой из песен второго поколения, и так далее.

Output

Так как ответ на задачу может быть достаточно большим, посчитайте его в виде несократимой рациональной дроби $\frac{A}{B}$ и выведите значение $(A \cdot B^{-1}) \bmod (10^9 + 7)$. Здесь B^{-1} — обратное к числу B по модулю $10^9 + 7$. Ограничения на входные данные гарантируют, что знаменатель дроби B не будет делиться на $10^9 + 7$, поэтому это выражение корректно определено.

Examples

stdin	stdout
2 2 1 1 1 2 1 1 1	666666672
2 2 1 1 1 2 1 1 2	520833338
2 2 1 1 2 2 1 1 1	916666674

Note

В первом тесте из примера оба поколения порождают от 0 до 2 песен с вероятностями $\frac{1}{3}$. В третьем поколении будет:

- 0 песен с вероятностью $\frac{13}{27}$,
- 1 песня с вероятностью $\frac{5}{27}$,
- 2 песни с вероятностью $\frac{2}{9}$,
- 3 песни с вероятностью $\frac{2}{27}$,
- 4 песни с вероятностью $\frac{1}{27}$.

В таком случае искомое математическое ожидание равно $\frac{2}{9} \binom{2}{2} + \frac{2}{27} \binom{3}{2} + \frac{1}{27} \binom{4}{2} = \frac{2}{3}$.

На второй тест ответ $\frac{49}{48}$, на третий — $\frac{11}{12}$.