

## Задача А. Сортировка медианой

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

Это интерактивная задача.

Программа жюри загадала перестановку  $P$  чисел от 1 до  $n$ :  $P = \langle P_1, P_2, \dots, P_n \rangle$ . Вашей программе требуется её отгадать.

Вы можете задавать следующие вопросы: по заданным трем различным позициям  $i$ ,  $j$  и  $k$ , выясните, какое из чисел  $P_i$ ,  $P_j$  и  $P_k$  является медианой? Медиана — элемент, отличный от минимального и максимального.

Например, пусть была загадана перестановка  $\langle 2, 1, 4, 3 \rangle$ , и ваша программа задает вопрос про позиции 1, 2 и 3, ответом будет 2, поскольку 2 — медиана множества  $\{P_1, P_2, P_3\} = \{2, 1, 4\}$ . Обратите внимание, что вашей программе не сообщается, на какой из позиций 1, 2 и 3 находится медиана.

Хотелось бы найти перестановку  $P$ . К сожалению, её невозможно определить однозначно, для каждой перестановки  $P$  обозначим как  $S(P)$  множество перестановок, которые невозможно отличить от  $P$  с помощью вопросов указанного типа. Вам требуется найти любую перестановку из множества  $S(P)$ .

### Протокол взаимодействия

Сначала требуется считать со стандартного ввода числа  $n$ , размер перестановки ( $3 \leq n \leq 200$ ). Затем программа может задавать вопросы указанного типа, выводя на стандартный вывод строку, содержащую три различных числа от 1 до  $n$ . После каждого вопроса ваша программа должна считать со стандартного ввода одно число — медиану значений на соответствующих позициях. Вы можете задавать вопросы, пока не определите перестановку из множества  $S(P)$ . Разрешается задать не более 2000 вопросов.

Когда ваша программа найдет ответ, выведите слово “OK”, и зачем числа  $P_1, P_2, \dots, P_n$ . После этого ваша программа должна выйти.

### Пример

стандартный ввод	стандартный вывод
4	1 2 3
2	2 3 4
3	1 2 4
2	1 3 4
3	OK 2 1 4 3

## Задача В. Экспедиция Деда Мороза

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

Вы возглавляете экспедицию Деда Мороза по доставке подарков детям в нескольких деревнях в Африке. Всего вы должны доставить подарки в  $n$  деревень, расположенных вдоль прямой. Утром первого дня ваша экспедиция находится в деревне номер 1.

В  $i$ -й деревне находится  $a_i$  детей. Каждый день вы можете сделать одно из двух. Либо подарить подарки всем детям в деревне, в которой находится экспедиция, либо переехать в соседнюю деревню. Если в конце какого либо дня в деревне подарки не доставлены, то дети в этой деревне плачут всю ночь.

Вы бы хотели доставить подарки детям во всех деревнях, чтобы суммарно было как можно меньше случаев, чтобы ребенок плакал ночью. Но есть одно ограничение, обусловленное тем, что вы не хотите, чтобы дети теряли надежду. Пусть вы вошли в деревню  $i$  и не стали на следующий день дарить подарки детям в этой деревне, а перешли в другую деревню. Тогда затем, когда вы будете впервые перемещаться из деревни  $j$  в такую деревню  $k$ , что  $k$  ближе к  $i$ , чем  $j$ , то вы после этого должны перемещаться (возможно, раздавая подарки детям по дороге), не меняя направления, в сторону деревни  $i$  и затем раздать там подарки детям.

Например, пусть вы исходно в деревне 1. В первый день вы переезжаете в деревню 2. Во второй день вы переезжаете в деревню 3. В третий день вы дарите подарки детям в деревне 3. На следующий день вы переезжаете в деревню 2. Теперь по правилу из предыдущего абзаца вы должны на следующий день подарить подарки детям в деревне 2. Затем вы обязаны переехать в деревню 1 и подарить подарки детям там. Далее ограничения заканчиваются.

### Формат входных данных

Входные данные содержат несколько тестовых наборов.

Первая строка каждого тестового набора содержит  $n$  — число деревень ( $1 \leq n \leq 3000$ ). Следующая строка содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ). После последнего тестового набора идет строка, содержащая  $n = 0$ .

Сумма всех  $n$  в одних входных данных не превышает 3000.

### Формат выходных данных

Для каждого тестового набора выведите одно число: минимальное возможное суммарное число пар ребенок-ночь, что ребенок плакал в эту ночь.

### Пример

стандартный ввод	стандартный вывод
6 40 200 1 300 2 10 0	1950

### Замечание

В приведенном примере оптимальный порядок действий следующий: пойти в деревню 2, подарить подарки в деревне 2, пойти в деревню 3, пойти в деревню 4, подарить подарки в деревне 4, пойти в деревню 3, подарить подарки в деревне 3 (обязательно), подарить подарки в деревне 2 (обязательно), пойти в деревню 1 (обязательно), подарить подарки в деревне 1 (обязательно), пойти в деревни 2, 3, 4, 5, 6, соответственно, подарить подарки в деревне 6, пойти в деревню 5, подарить подарки в деревне 5 (обязательно).

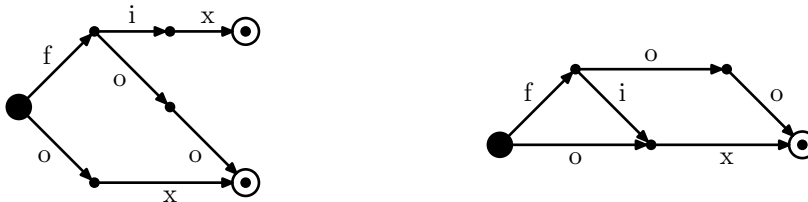
Экспедиция занимает 18 дней, в каждый из этих день плачет ночью, соответственно, 553, 353, 353, 353, 53, 53, 52, 52, 52, 12, 12, 12, 12, 12, 12, 2, 2, 0 детей.

## Задача С. Распознавание формального языка

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

Детерминированный конечный автомат представляет собой ориентированный граф, в котором может быть несколько ребер между парой вершин. Каждое ребро помечено одним из символов алфавита. Вершины графа — состояния автомата, а ребра — переходы. Для каждого состояния и каждого символа из этого состояния выходит не более одного ребра, помеченного этим символом. Одно из состояний автомата помечено как *стартовое*, а некоторые состояния помечены как *терминальные*. Говорят, что слово допускается автоматом, если начав со стартового состояния и проходя по очереди по ребрам, помеченным символами этого слова, в конце автомат попадает в терминальное состояние. *Языком автомата* называется множество слов, которые допускаются заданным автоматом.

Если задано конечное множество слов, то всегда можно построить автомат, допускающий в точности это множество слов. Например, картинка слева показывает автомат для множества слов *fix*, *foo*, *ox*. В этом автомате 7 состояний, это не оптимально. На картинке справа показан автомат для того же языка, содержащий лишь 5 состояний. Это количество является минимальным возможным.



Ваша задача — по заданному множеству слов найти минимальное число состояний в автомате, который допускает в точности заданное множество.

### Формат входных данных

Первая строка ввода содержит число  $n$  ( $1 \leq n \leq 5000$ ) — число слов в заданном множестве. Далее следует  $n$  слов, по одному на строке. Каждое слово содержит от 1 до 30 строчных букв латинского алфавита (от “a” до “z”). Все слова различны.

### Формат выходных данных

Выведите одно целое число — минимальное число состояний в автомате, который допускает в точности заданное множество слов.

### Примеры

стандартный ввод	стандартный вывод
3 fix foo ox	5
4 a ab ac ad	3

## Задача D. Разбиения, избегающие подмножеств

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

Разбиением множества  $S = \{1, 2, \dots, n\}$  называется набор  $\mathcal{P}$  множеств  $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ , таких что их объединение  $\bigcup P_i = S$ , и никакие два из них не пересекаются:  $P_i \cap P_j = \emptyset$  для  $i \neq j$ . Примером разбиения для  $n = 5$  является  $P_1 = \{1, 3\}$ ,  $P_2 = \{2, 4, 5\}$ .

Будем говорить, что разбиение *избегает подмножества*  $Q$ , если ни одно из  $P_i$  не содержит  $Q$  в качестве подмножества. Например, приведенное в предыдущем абзаце разбиение избегает подмножеств  $\{1, 2\}$  и  $\{3, 4\}$ , но не избегает  $\{1, 3\}$  или  $\{2\}$ .

По заданному  $n$  и набору множеств  $Q_1, Q_2, \dots, Q_l$  найдите количество разбиений  $S$ , которые избегают каждого из  $Q_i$ . Выведите ответ по модулю  $10^9 + 7$ .

### Формат входных данных

Первая строка ввода содержит два целых числа  $n$  и  $l$  ( $1 \leq n \leq 100$ ,  $0 \leq l \leq 10$ ). Следующие  $l$  строк описывают множества, которых требуется избегать. Каждая строка начинается с числа  $q_i$  — размера множества, затем следует  $q_i$  чисел — элементы множества.

### Формат выходных данных

Выведите одно число — количество разбиений  $S$ , которые избегают каждого из  $Q_i$ , взятое по модулю  $10^9 + 7$ .

### Пример

стандартный ввод	стандартный вывод
5 2 3 1 2 3 2 2 4	34