Задача A. Конфеты Кирилла

Имя входного файла:	`combination.in`
Имя выходного файла:	`combination.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

У Кирилла из параллели C было k конфет, и он захотел их раздать ученикам своей параллели. Однако заметил, что конфет у него меньше чем учеников в параллели. Кирилл сел на скамейку и задумался. Просидев полчаса и доев последнюю конфету он подумал — интересно, а сколько было способов раздать все k конфет n ученикам параллели C, если конфеты нельзя делить, а каждому школьнику можно дать не более одной конфеты.

Формат входных данных

В единственной строке записаны числа n, $k (1 \leqslant k \leqslant n \leqslant 64)$.

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

combination.in	combination.out
5 3	10

Задача B. Тасовка

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Тасование колоды карт происходит следующим образом. Колода разбивается на несколько частей перегородками, которые нумеруются по номеру стоящей после неё карты (карты нумеруются с нуля). Затем полученные таким образом блоки карт переставляются в обратном порядке (при этом внутри каждого блока, порядок карт сохраняется). Требуется по заданным значениям карт и заданными номерами перегородок, вывести порядок карт после одного такого тасования.

Формат входных данных

Во входном файле находятся две строки. В первой строке содержатся значения карт по порядку через пробел. При этом гарантируется, что значение каждой карты по модулю не более 10^{15}, а количество карт не более 10^6. Во второй строке содержатся номера перегородок в порядке возрастания через пробел. Перегородки могут ставиться только между картами. Ни в каком промежутке между картами не может находиться более одной перегородки.

Формат выходных данных

В выходной файл надо вывести одну строку — полученную в результате тасовки последовательность значений карт.

Примеры

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

Задача C. Перемешиватель

Имя входного файла:	`permutator.in`
Имя выходного файла:	`permutator.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Пете надоело перемешивать карты для настольных игр перед каждой игрой, и он заказал по интернету специальный автоматический перемешиватель. Перемешиватель принимает на вход стопку из n карт и возвращает ее в перемешанном виде.

Вскоре друзья Пети заметили, что перемешиватель меняет порядок карт всегда одинаково. Карта, стоящая на месте i после перемешивания оказывается на месте p_i. Поэтому было решено пропускать карты через перемешиватель k раз, чтобы никто из игроков не мог проследить, какая карта попала на какое место.

Однако Петя решил, что особенностью аппарата надо воспользоваться и хочет написать программу, которая посчитает, на каком месте окажется каждая карта после k перемешиваний.

Формат входных данных

Первая строка входного файла содержит целые числа n и k ($1 \leqslant n \leqslant 1000$, $1 \leqslant k \leqslant 1000$). Следующая строка содержит n чисел p_i ($1 \leqslant p_i \leqslant n$, все p_i различны).

Формат выходных данных

Выведите в выходной файл n чисел. i-е число должно быть равно финальной позиции карты, которая была в колоде на i месте до перемешивания.

Примеры

permutator.in	permutator.out
5 2 2 3 1 5 4	3 1 2 4 5

Задача D. Перемешиватель 2

Имя входного файла:	`permutator2.in`
Имя выходного файла:	`permutator2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Друзья Пети заметили, что он слишком часто выигрывает и догадались, что он написал решение предыдущей задачи. Тогда они решили увеличить ограничение на число перемешиваний до 10^{18}, чтобы программа Пети не успевала вычислять ответ.

Помогите Пете решить задачу для больших значений k.

Формат входных и выходных данных тот же, что и в предыдущей задаче. Пример тоже можно посмотреть там. Единственное отличие этой задачи — ограничение на $k \leqslant 10^{18}$.

Задача E. Перемешиватель 3

Имя входного файла:	`permutator3.in`
Имя выходного файла:	`permutator3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Друзья Пети поняли, что он опять мухлюет с перемешивателем и решили его проучить. Они хотят подобрать такое k, что после запуска перемешивателя раз k раз, карты встают на те же места, на которых были до перемешивания. Помогите друзьям найти минимальное такое k.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leqslant n \leqslant 1000, 1 \leqslant k \leqslant 1000$). Следующая строка содержит n чисел p_i — перестановку, которую производит перемешиватель.

Формат выходных данных

Выведите в выходной файл минимальное число раз, которое нужно запустить перемешиватель, чтобы все карты встали на исходные места.

Примеры

permutator3.in	permutator3.out
5 2 3 1 5 4	6

Задача F. Спор о математике

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

— А я говорю, ком-мур-тирует!

— Ни в коем случае! Это чистейшей воды провокация...

Спор о математике был не менее жарким, чем подобные баталии свиты на литературную тему. Впрочем, на этот раз существовал абсолютно точный ответ.

— Азазелло, не шуми. Лучше покажи Коту, что не коммутирует, и поскорее. Первая перестановка записана на корешке книги.

— Слушаюсь, мессир.

Азазелло еле успел прочесть набор чисел, после чего книга буквально сгнила в его руках. У него оставалась всего лишь секунда, чтобы выписать перестановку, не коммутирующую с данной. Справитесь ли вы?

Введем немного определений.

Перестановкой называется последовательность $\{p_i\}$ из n натуральных чисел от 1 до n, каждое из которых встречается ровно один раз.

Композицией перестановок p и q называется перестановка $pq = z$, определяемая так: $z_i = p_{q_i}$.

Две перестановки p и q *коммутируют*, если $pq = qp$.

Формат входных данных

Входные данные содержат описания нескольких перестановок.

В первой строке каждого описания записано число n — количество элементов в перестановке ($1 \leqslant n \leqslant 100\,000$). Во второй строке записано n различных целых чисел от 1 до n — исходная перестановка.

Гарантируется, что сумма n по всем перестановкам во вводе не превосходит 10^5.

Формат выходных данных

Для каждой перестановки необходимо вывести одну или две строки. Если существует перестановка, не коммутирующая с данной, выведите в первой строке слово `Yes`, а во второй строке n чисел — любую такую перестановку. В противном случае выведите единственную строку со словом `No`.

Примеры

commute.in	commute.out
3 2 1 3 2 1 2	Yes 2 3 1 No