Задача A. Красивые числа

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

Назовем число *красивым*, если сумма квадратов его цифр в десятичной системе счисления является простым числом. Найдите n в порядке возрастания красивое число.

Напомним, что положительное целое число x называется *простым*, если оно больше единицы и не делится ни на одно из чисел от 2 до $x - 1$.

Формат входного файла

Во входном файле находится положительное целое число n ($1 \le n \le 10000$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

beauty.in	beauty.out
1	11
6	23

Задача B. Планета Константинов

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

Как известно, семьи на планете Константинов очень большие. Не менее известный факт —- то, что фантазия у константинян (обитателей планеты Костантинов) весьма и весьма скудна. За долгие тысячелетия существования они не придумали и миллиона-другого имен... В связи с этим обстоятельством, чтобы не путать друг друга, у среднестатистического Костантина с планеты Константинов может быть не одно, не десять, а целая сотня имен (разумеется, помимо основного имени Константин, которое в повседневной жизни опускается), идущих в строго зафиксированном порядке. Перепутать порядок следования имен при обращении к константинянину —- верный способ стать его заклятым врагом. В истории планеты Константинов известны даже случаи казни тех, кого угораздило неправильно обратиться к венценосной особе.

К сожаления, про планету Константинов известно не так много. В частности, пока в руки ученых попал список детей одной семьи. Известно, что имя того, кто составлял этот список, присутствует в нем. Кроме того, автор подписался в самом низу списка. Согласно древней традиции, когда константинянин подписывается, он не должен отрывать пера от бумаги, соответственно все пробелы опускаются. Ученые хотели выяснить, кто же составил этот список. Однако эта задача оказалась не из легких, так как возможных авторов оказалось много. Например, если подписался некто «Тридцатьтрипопугая», то это мог быть как «Три Дцать Трипоп Угая», так и «Тридц Атьт Рипопуг А Я»

Столкнувшись с этой проблемой, ученые решили для начала узнать количество обитателей планеты Костантинов, которые есть в списке и подпись которых совпадает с подписью автора списка.

Формат входного файла

Во входном файле дано число n ($1 \leq n \leq 1000$) —- количество имен в списке. Далее следует n строк, в каждой из которых содержатся в правильном порядке имена константинян, разделенные произвольным числом пробелов. Имена состоят из прописных латинских букв. На последней строке входного файла содержится подпись автора списка. Длина каждой строки во входном файле не превосходит 1024 символов.

Формат выходного файла

В выходной файл выведите количество константинян, которые есть в списке и подпись которых совпадает с подписью автора списка.

Примеры

constant.in	constant.out

`1`
`constantine`
`constantine`
	`1`

`3`
`one two three`
`on et woth ree`
`onetwoth tree`
`onetwothree`
	`2`

Задача C. Выпуклый?

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

Назовем многоугольник *выпуклым*, если все его вершины лежат в одной полуплоскости относительно любой прямой, проходящей через его сторону. Задан некоторый многоугольник. Необходимо проверить, является ли он выпуклым.

Дадим определение полуплоскости. Пусть прямая задана уравнением $Ax+By+C=0$. Полуплоскостью называется геометрическое место точек удовлетворяющее неравенству $Ax+By+C \geq 0$ (либо неравенству $Ax + By + C \leq 0$).

Формат входного файла

В первой строке входного файла записано число n ($4 \leq n \leq 10^3$). Каждая из следующих n строк содержит два целых числа — координаты соответствующей вершины n-угольника. Все координаты во входном файле целые и не превышают 10^8 по абсолютному значению.

Вершины многоугольника заданы в порядке обхода против часовой стрелки. Многоугольник не имеет самопересечений и самокасаний, никакие три последовательные вершины не лежат на одной прямой.

Формат выходного файла

В выходной файл выведите слово `YES`, если многоугольник является выпуклым, и слово `NO` — иначе.

Примеры

convex.in	convex.out
4 0 0 1 0 1 1 0 1	YES
4 0 0 3 0 3 3 2 1	NO

Задача D. Циклический класс

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

Перестановкой порядка n называется последовательность из попарно различных целых положительных чисел $p_1, p_2, \ldots, p_n$, где каждое $1 \le p_i \le n$. *Циклом перестановки* называется такая последовательность $i_1, i_2, \ldots, i_k$ различных положительных целых чисел от 1 до n, что $p_{i_1} = i_2$, $p_{i_2} = i_3, \ldots, p_{i_k} = i_1$.

Любую перестановку можно единственным образом разбить на несколько циклов так, чтобы каждый ее элемент принадлежал ровно одному циклу. *Циклическим классом* перестановки называется последовательность, состоящая из длин ее циклов, упорядоченная по неубыванию.

Задана перестановка. Найдите ее циклический класс.

Формат входного файла

Первая строка входного файла содержит порядок n ($1 \le n \le 1000$) заданной перестановки. Вторая строка содержит числа $p_1, p_2, \ldots, p_n$, отделенные друг от друга пробелами.

Формат выходного файла

В первой строке выходного файла выведите количество циклов c в заданной перестановке. Во второй строке выведите длины циклов в порядке неубывания.

Примеры

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

Задача E. Сумма k подмножеств

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

Одной из классических NP-полных задач является так называемая *задача о сумме подмножества* (SUBSET-SUM problem). Она формулируется следующим образом: «Дано n чисел a_1, a_2, ..., a_n. Возможно ли выбрать несколько из них так, чтобы их сумма была равна S?».

Эта задача допускает несколько обобщений. Одно из них мы назовем *задача о сумме k подмножеств* и будем рассматривать далее. Пусть задано n чисел a_1, a_2, ..., a_n и k чисел s_1, s_2, ..., s_k. Необходимо разбить числа a_1, ..., a_n на k непустых наборов, чтобы сумма чисел в первом из них была равна s_1, во втором — s_2, ..., в k-ом — s_k. Каждое число может включено не более чем в один набор.

Напишите программу, решающую задачу о сумме k подмножеств.

Формат входного файла

Первая строка входного файла содержит два целых числа n и k ($1 \le n, k \le 10$, $n^k \le 2^{24}$). Вторая строка входного файла содержит n целых чисел a_1, a_2, ..., a_n (для всех a_i верно неравенство $|a_i| \le 10^9$). Третья строка входного файла содержит k целых чисел s_1, s_2, ..., s_k (для всех s_j верно неравенство $|s_j| \le 10^9$).

Формат выходного файла

Если искомое разбиение на наборы существует, то выведите в первой строке выходного файла слово `YES`, иначе — слово `NO`.

Примеры

ksubset.in	ksubset.out
4 2 1 2 3 4 5 5	YES
4 2 1 2 3 4 1 8	NO

Задача F. Путь из S в T

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

Вы являетесь одним из работников отдела логистики одной крупной транснациональной корпорации. Недавно ваша корпорация начала осваивать рынок страны N-мерики. Первый филиал открылся в городе S. Ближайшая цель — открыть торговое представительство в городе T. Для этого туда необходимо перевести оборудование суммарной массой M тонн. Оборудование упаковано в контейнер, который запрещается вскрывать до прибытия в город T, поэтому его придется перевозить как единое целое.

Известна карта дорог N-мерики и максимальный тоннаж груза, который разрешено перевозить по каждой из дорог. Необходимо выяснить, можно ли перевезти груз массой M тонн из города S в город T.

Формат входного файла

В первой строке входного файла находятся два целых числа n и m — количество городов и дорог в N-мерике ($2 \le n \le 1000$, $0 \le m \le 100000$). Города пронумерованы натуральными числами от 1 до n, город S имеет номер 1, город T — номер n. Вторая строка входного файла содержит целое число M ($1 \le M \le 10^9$).

Последующие m строк описывают каждая одну дорогу и содержат по три целых числа: u, v, w — соответственно, номера городов, которые соединены этой дорогой ($1 \le u \ne v \le n$) и максимальный вес груза, который можно перевозить по этой дороге ($1 \le w \le 10^9$). По дорогам разрешено движение в обе стороны, любые два города соединены не более чем одной дорогой.

Формат выходного файла

В выходной файл выведите слово `YES`, если указанный груз можно перевезти из города S в город T, и слово `NO` — в противном случае.

Примеры

path.in	path.out
4 4 5 1 2 5 2 3 6 2 4 4 3 4 7	YES
4 4 5 1 2 5 2 3 4 2 4 4 3 4 7	NO

Задача G. Шифр «Змейка»

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

Один из способов шифрования, который часто используют начинающие криптографы, заключается в том, что шифруемое слово записывается некоторым достаточно хитрым образом в клетках квадрата. Одним из таких, не очень удачных шифров, является шифр «Змейка».

Он применяется для шифрования слов длины n^2 (если длины слова меньше, оно дополняется точками). Слово разбивается на части длиной по n символов каждая. Далее эти части записываются в квадрат размером n на n (каждая часть в отдельную строку), при этом части с четными номерами записываются задом наперед. Например, слово `Iloveyou.`, полученное из фразы `I love you.` удалением пробелов, будет закодировано следующим образом.

Оно будет разбито на три части: `Ilo`, `vey`, `ou.`. Эти части далее будут записаны в квадрат так:

I	l	o
y	e	v
o	u	.

Можно видеть, что шифруемое слово записано в этом квадрате «змейкой».

Задан квадрат, получившийся в результате шифрования некоторого слова описанным способом. Необходимо выяснить, содержится ли в исходном слове подстрока t.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \le n \le 50$). Последующие n строк содержат каждая по n символов — строки квадрата, начиная с первой. Последняя строка входного файла содержит искомую непустую строку t, длина которой не превышает 1000 символов. Квадрат и строка t состоят только из строчных букв латинского алфавита и точек.

Формат выходного файла

В выходной файл выведите слово `YES`, если строка t содержится в зашифрованной строке, и слово `NO` — в противном случае.

Примеры

snake.in	snake.out
3 Ilo yev ou. you	YES
3 Ilo yev ou. me	NO

Задача H. Две последовательности

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

Циклический преобразователь последовательностей (ЦПП) — это специальное устройство, работающее с последовательностями из чисел от 0 до $k-1$. Пусть задана последовательность a_1, a_2, $\ldots$, a_n. За одну операцию ЦПП может заменить a_i на $(a_i + d) \bmod k$ для всех $1 \le l \le i \le r \le n$. Такая операция в дальнейшем будет записываться как `l r d`.

Одной из интересных задач, которую может решать ЦПП, является преобразование одной последовательности в другую. Пусть заданы две последовательности a_1, a_2, $\ldots$, a_n и b_1, b_2, $\ldots$, b_n чисел от 0 до $k-1$. Необходимо преобразовать первую из них во вторую за не более чем $n-1$ операцию.

Формат входного файла

Первая строка входного файла содержит два целых числа n и k ($1 \le k < n \le 1000$). Вторая строка входного файла содержит n целых чисел a_1, a_2, $\ldots$, a_n. Третья строка входного файла содержит n целых чисел b_1, b_2, $\ldots$, b_n. Для всех a_i и b_i верно неравенство $0 \le a_i, b_i < k$.

Формат выходного файла

В первой строке выходного файла выведите количество преобразований c ($0 \le c < n$, его не требуется минимизировать). В последующих c строках выведите описания этих преобразований в формате `l r d`. Числа d для каждой из операций должны быть в диапазоне от нуля до $k-1$ включительно.

Примеры

twoseq.in	twoseq.out
5 3 0 0 1 2 2 1 1 2 0 0	3 1 2 1 3 3 1 4 5 1