Об уровнях олимпиады

В этой интернет-олимпиады мы проводим эксперимент с уровнями сложности.

Вы можете самостоятельно выбрать, задачи какого уровня решать.

Если вы хотите участвовать в олимпиаде базового уровня, решайте задачи A, B, C и D.

Если вы хотите участвовать в олимпиаде усложненного уровня, решайте задачи C, D, E и F.

У базового и усложненного уровней будут отдельные таблицы результатов.

Если вы отправите решение задачи E или F, вы автоматически будете классифицированы в таблице усложненного уровня (даже если ваше решение не пройдет тесты из примера). Иначе вы будете классифицированы в таблице базового уровня.

Задача A. Диагональное преобладание

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

В компании *Macrohard* изучают матрицы с диагональным преобладанием, так необходимые для написания новой операционной системы *Doors 8*. Матрица A размером $n \cdot n$ называется матрицей с *диагональным преобладанием*, если выполнены следующие три условия:

1. $A_{i,j} \geq 0$ $(1 \leq i, j \leq n)$
2. $A_{i,i} \geq \sum\limits_{j=1}^{i-1} A_{i,j} + \sum\limits_{j=i+1}^{n} A_{i,j}$ $(1 \leq i \leq n)$
3. Найдется i, такое что $A_{i,i} > \sum\limits_{j=1}^{i-1} A_{i,j} + \sum\limits_{j=i+1}^{n} A_{i,j}$

Диагональный элемент матрицы $A_{i,i}$ называется преобладающим, если

- $A_{i,i} > \sum\limits_{j=1}^{i-1} A_{i,j} + \sum\limits_{j=i+1}^{n} A_{i,j}$

До недавнего времени сотрудники *Macrohard* исследовали матрицы сами, используя калькулятор. Однако вчера они устроили забастовку, так как устали делать одни и те же операции с большими матрицами. Вас нанял главный начальник *Macrohard* и попросил написать программу, автоматизирующую громоздкие подсчёты работников. Не оплошайте!

Формат входного файла

В первой строке входного файла дано число n $(2 \leq n \leq 1000)$ — размер матрицы. Далее следует n строк, в каждой из которых записано n чисел — элементы матрицы $(0 \leq A_{i,j} \leq 1000)$.

Формат выходного файла

В выходной файл выведите "`YES`", если матрица является красивой, "`NO`" в противном случае. Также при условии диагонального преобладания выведите на следующей строчке количество её преобладающих элементов.

Примеры

diagonal.in	diagonal.out
10 5 5 0 1 0 2 2 5	YES 2
10 5 5 0 1 0 2 2 3	NO

Задача B. Соло на клавиатуре

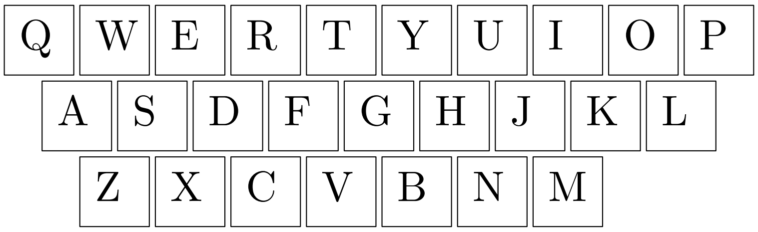
Имя входного файла:	`keyboard.in`
Имя выходного файла:	`keyboard.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Работники *Министерства Быстрой Печати и Длинных Строк* озабочены новой проблемой. В последнее время сложность набираемых ими текстов существенно увеличилась, и они стали допускать ошибки. К счастью, уровень подготовки в *МБПДС* высокий, поэтому любой работник в любом слове допускает не более одной ошибки.

Ошибки, которые могут допустить работники *МБПДС*, таковы:

- вместо нужной клавиши нажата соседняя;
- клавиша не нажата;
- перед нужной клавишей нажата одна из соседних клавиш;
- после нужной клавиши нажата одна из соседних клавиш;

Для клавиши на клавиатуре соседними считаются шесть клавиш: слева, справа, слева сверху, справа сверху, слева снизу, справа снизу. Клавиатура является стандартной QWERTY-клавиатурой и изображена на рисунке.



Министерство хочет автоматизировать процесс исправления ошибок. Для этого нужно написать программу, которая могла бы по данному слову, в котором допущено не более одной ошибки, выводить список словарных слов, которые могли быть введены.

Помогите работникам *МБПДС* и напишите такую программу.

Формат входного файла

В первой строке входного файла находится число N $(1 \le N \le 5 \cdot 10^4)$ — число слов в словаре. В каждой из следующих N строк содержится по одному слову из словаря. Все слова в словаре различны.

В следующей строке находится число M $(1 \le M \le 5 \cdot 10^4)$ — слова, которые необходимо исправить. Каждая из следующих M строк содержит по одному слову для проверки.

Все слова в файле состоят из прописных латинских букв. Во входном файле не содержится пустых строк. Размер входного файла не превышает одного мегабайта.

Формат выходного файла

Для каждого из слов, подлежащих исправлению, выведите информацию о возможных его исправлениях. Эта информация состоит из блока, включающего одну или несколько строк. Первая строка блока содержит одно целое число C_i — число словарных слов, которые могли бы соответствовать введенному слову. Далее необходимо вывести эти словарные слова в лексикографическом порядке, по одному на каждой строке. Если $C_i > 5$, выведите только первые пять слов.

Примеры

keyboard.in	keyboard.out
7 solving contests is what makes us happy 3 solvung js qwerty	1 solving 2 is us 0
6 letterq letterw lettere letterr lettert lettery 2 letter lertterq	6 lettere letterq letterr lettert letterw 1 letterq

Пусть K — максимальная длина слова во входном файле.
Решения для $N, M \le 500$, $K \le 20$ будут оцениваться из 50 баллов.
Решения для $K \le 100$ будут оцениваться из 80 баллов.

Задача C. Лотерея

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

Планета Шелезяка. Воды нет. Полезных ископаемых нет. Населена роботами.

Недавно на планету Шелезяка прибыла новая партия чистейшей смазки. К сожалению, на всех ее не хватает. Поэтому было решено провести лотерею. Для этого было выбрано n натуральных чисел $a_1, a_2 \ldots a_n$ и число k, $0 \leq k \leq n$. Смазку получат те роботы, серийные номера которых делятся ровно на k из этих чисел.

Теперь правительство заинтересовало то, насколько равномерно распределены призы. Для этого они решили выяснить, какое число роботов, серийные номера которых принадлежат промежутку $[a..b]$, получат смазку.

Формат входного файла

В первой строке входного файла заданы натуральные числа n ($1 \leq n \leq 10^5$) и k ($0 \leq k \leq n$). Во второй строке входного файла перечислены n натуральных $a_1, a_2 \ldots a_n$, разделенные пробелами. Каждое из этих a_i не превосходит 10^9. В третьей строке входного файла задано два натуральных числа a и b ($1 \leq a \leq b \leq 10^9$, $b - a \leq 10^5$).

Формат выходного файла

В выходной файл выведите количество чисел из промежутка $[a..b]$, которые делятся ровно на k данных чисел.

Примеры

lottery.in	lottery.out
2 1 2 3 1 10	6
4 2 2 3 4 5 10 30	5

Задача D. Возведение в степень

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

Первокласснику Васе родители подарили на день рождения калькулятор. Вася быстро изучил все возможности своего нового калькулятора. Больше всего ему понравилась операция возведения в степень. Он был очень впечатлен тем, что даже если небольшое число возвести в степень с небольшим показателем, то результат может получиться очень большой.

Возводить в степень сам Вася пока не умеет, но очень хочет научиться. Для начала он решил научиться определять по числам a и b, сколько цифр будет в десятичной записи числа a^b (a в степени b). Однако это у него не получилось, и он обратился за помощью к своему другу, второкласснику Роме. Рома — большой любитель делать все наоборот. Он предложил Васе сначала научиться по числу k находить числа a и b, такие что число a^b состоит из k цифр в десятичной записи. Поскольку таких a и b может быть большое количество, Вася попытался научиться выяснять, сколько существует пар (a, b), удовлетворяющих данному условию.

К сожалению, Васе не удалось и это, и теперь он просит о помощи Вас. Напишите программу, отвечающую на поставленную задачу.

Формат входного файла

Во входном файле задано единственное целое положительное число k ($1 \le k \le 17$).

Формат выходного файла

В выходной файл выведите количество различных пар (a, b), таких что a и b — целые положительные числа ($a > 1$), и число a^b состоит из k цифр в десятичной записи.

Примеры

pow.in	pow.out
2	102
3	934

Задача E. Слепые флибы

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

Флибы — фантастические существа, которые предсказывают будущее. Флибы часто используются при демонстрации использования генетических алгоритмов. В этой задаче мы исследуем свойства «слепых» флибов, которые пытаются предсказывать будущее, не получая никакой информации из окружающего мира.

Представим будущее в виде строки w, состоящей из нулей и единиц, приписанную к самой себе до бесконечности. В результате получается бесконечная строка w^*. Например, если $w = 010011$, будущее представляет собой строку $w^* = 010011010011010011\ldots$. Слепой флиб — детерминированный конечный автомат, который выводит символ на каждом переходе.

Формально слепой флиб имеет n состояний, из каждого состояния есть переход, помеченный 0 и переход, помеченный 1. Переход — это тройка $\langle m, p, s \rangle$, где m — символ, которым помечен переход, p — символ, который выводится на переходе, а s — состояние, в которое флиб переходит, когда происходит этот переход.

Исходно флиб выводит некоторый символ (он может решить, какой) и переходит в состояние 1. После этого флиб действует следующим образом. Пусть c — последний символ, который вывел флиб. Он выбирает переход, помеченный символом c, и переходит по нему (выводя соответствующий этому переходу символ). Этот процесс продолжается до бесконечности.

Пусть z — бесконечная строка, выведенная флибом. Обозначим как $e(k)$ количество позиций среди первых k, на которых в строках z и w^* стоят одинаковые символы. При увеличении k отношение $\frac{e(n)}{n}$ становится сколь угодно близко к некоторому числу P. Это число называется *предсказательной способностью* флиба. Чем выше предсказательная способность — тем лучше флиб.

Например, очевидно, для любого w существует слепой флиб с предсказательной способностью 1.0? содержащий n состояний, где n — длина строка w. Состояние такого флиба соответствует позиции в строке w, до которой к текущему моменту она была выведена. Он всегда правильно предсказывает следующий символ.

Задано слово w. Для каждого k от 1 до длины w постройте слепого флиба с максимальной предсказательной способностью, имеющего k состояний.

Формат входного файла

Входной файл содержит слово w (длина w от 1 до 200, включительно, w содержит только 0 и 1).

Формат выходного файла

Для всех k от 1 до длины w выведите максимальную возможную предсказательную способность слепого флиба с k состояниями.

Примеры

blind.in	blind.out

`010011`	`0.6666666666666667`
	`0.8333333333333334`
	`1.0`
	`1.0`
	`1.0`
	`1.0`

Задача F. Покрытие строки

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

Будем говорить, что строка α покрывает строку β, если для каждой позиции строки β найдется такое вхождение α в β как подстроки, которое содержит эту позицию. Например, строка "`aba`" покрывает строку "`abaabaababa`", но не покрывает строку "`baba`". Очевидно, любая строка покрывает сама себя.

Задана строка w. Для каждого ее префикса $w[1..k]$ найдите самую короткую строку, которая покрывает этот префикс.

Формат входного файла

Входной файл содержит строку w, состоящую из строчных букв латинского алфавита. Длина строки w не превышает 250 000.

Формат выходного файла

Для каждого k от 1 до длины w выведите длину самой короткой строки, которая покрывает $w[1..k]$.

Примеры

cover.in	cover.out
abaabaababa	1 2 3 4 5 3 4 5 3 10 3