Задача A. Криптография

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

Кристине очень нравится заниматься в математическом кружке. На последнем занятии из небольшой лекции по криптографии она узнала, что столь популярный подстановочный шифр, на самом деле, очень легко вскрывается. Кристина поспорила с подругами, что сможет прочитать любую записку, которую они зашифруют. Правда она не хочет заниматься рутинной работой, тем более, что при этом можно еще и сделать ошибку. Вы ведь поможете написать программу, которая справится с расшифровкой?

Кристина снабдила вас следующими сведениями. *Подстановочным* называется шифр, при использовании которого каждая буква сообщения заменяется какой-нибудь другой буквой, а все пробелы и знаки препинания удаляются. При этом для сохранения возможности однозначной расшифровки никакие две буквы не превращаются в одну.

Зная тот факт, что частоты появления в осмысленном тексте букв различаются, можно с хорошей точностью установить, каким образом были зашифрованы буквы. Среди всех возможных способов перестановки букв следует выбрать такой, для которого минимально значение величины

$$\Sigma_c|e_c - p_c|$$

Здесь через e_c и p_c обозначены частоты символа c, рассчитанные для среднего текста и для записки, получающейся из зашифрованного текста после применения данной перестановки.

Кристина уже сосчитала частоты, с которыми встречаются в записках различные буквы, так что теперь осталось только написать программу.

Формат входного файла

В первой строке входного файла записаны два целых числа — n и m ($1 \le n \le 26, 1 \le m \le 100000$). Следующие n строк задают буквы, которые, по сведениям Кристины, могли встречаться в исходном тексте. Соответствующая буква c_i — первый символ строки, через пробел от него записана частота этой буквы p_i, заданная с 4 знаками после десятичной точки. Все p_i неотрицательны, а их сумма равна 1. Последняя строка содержит m символов — текст, который необходимо расшифровать.

Формат выходного файла

В выходной файл выведите n символов по одному на строке, которые, если верить описанному методу, должны быть сопоставлены символом из входного файла. Если оптимальных решений несколько, выведите любое из них.

Примеры

crypto.in	crypto.out
3 8 k 0.1250 e 0.6250 v 0.2500 wwpwopwo	p w o

Задача B. Сдача

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

Когда Миша и Маша покупали подарок, возникла интересная ситуация. У них была в распоряжении только одна большая купюра, а у продавца — некоторое количество мелочи. Дело происходило утром, поэтому продавцу нужно было экономить мелочь, и он хотел отдать сдачу минимальным количеством монет. Подумав некоторое время, они точно определили, с каким количеством монет продавцу придется расстаться.

А вы сможете решить такую задачу?

Формат входного файла

В первой строке входного файла записано число n ($1 \leq n \leq 10$) — количество различных номиналов монет, содержащихся в кассе. Можно считать, что количество монет каждого номинала достаточно.

На следующей строке содержится n целых чисел a_i ($0 < a_i \leq 2000$) — номиналы монет.

В третьей строке записано одно число k ($1 \leq k \leq 1000000$) — сумма, которую нужно набрать.

Формат выходного файла

В выходной файл выведите минимальное количество монет, которое придется отдать продавцу, или -1, если продавец вообще не сможет дать им сдачу.

Примеры

change.in	change.out
3 1 3 5 13	3
4 5 6 7 8 9	-1

Задача C. Секретное сообщение

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

На секретную базу в Арктике поступила шифровка — последовательность из n десятичных цифр. Она содержит номер секретной базы в Антарктиде, который является последовательностью из k десятичных цифр. При этом для того, чтобы отличить его от ненужной Вам информации, он повторен в шифровке хотя бы два раза (возможно, эти два вхождения перекрываются).

Напишите программу, которая по шифровке и длине номера секретной базы определяет, содержит ли шифровка номер базы. Учтите, что у базы может быть несколько номеров, и все они могут быть переданы в шифровке.

Формат входного файла

Первая строка входного файла содержит два целых числа: n $(1 \le n \le 100000)$ и k $(1 \le k \le 5)$ — длину шифровки и длину номера секретной базы соответственно. Вторая строка содержит n цифр — шифровку. Помните, что цифры в шифровке не разделяются пробелами.

Формат выходного файла

Выведите в выходной файл ‘`YES`’, если шифровка содержит номер секретной базы, и ‘`NO`’ в противном случае.

Примеры

cipher.in	cipher.out
10 5 0123456789	NO
13 2 0123400056789	YES

Задача D. Две окружности

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

На плоскости даны две окружности. Требуется проверить, пересекаются ли они.

Формат входного файла

Входной файл состоит из двух строк. На каждой строке записана информация об одной окружности — координаты ее центра x и y (целые числа, по модулю не превосходящие 5000) и радиус (целое число $1 \le r \le 5000$).

Формат выходного файла

Выведите в выходной файл '`YES`', если окружности пересекаются, и '`NO`' в противном случае.

Примеры

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

Задача E. Часы

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

Петя очень любит наблюдать за электронными часами. Он целыми днями смотрел на часы и считал, сколько раз встречается каждая цифра. Через несколько месяцев он научился по любому промежутку времени говорить, сколько раз на часах за это время встретится каждая цифра, и очень гордился этим.

Вася решил проверить Петю, но он не знает как решать эту задачу. Вася попросил Вас помочь ему. Напишите программу, решающую эту задачу.

Формат входного файла

Первая и вторая строки входного файла содержат начало и конец промежутка времени соответственно. Начальное время не превосходит конечное. Время задается в формате $hh:mm:ss$ ($0 \leq hh < 24$, $0 \leq mm < 60$, $0 \leq ss < 60$). hh, mm, ss дополнены ведущими нулями до двух символов. Эти нули также учитываются при подсчете числа цифр.

Формат выходного файла

Выходной файл должен содержать 10 строк. В i-ой строке должно быть написано, сколько раз встречается цифра $i-1$.

Примеры

clock.in	clock.out
23:59:58 23:59:59	0 0 2 2 0 4 0 0 1 3
13:24:09 13:24:40	5 45 45 45 36 3 3 3 3 4

Задача F. Письмо

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

Вася хочет отправить Пете письмо на листе шириной l. Он хочет распечатать на нём текст, состоящий из k строк. Необходимо изменить количество ведущих и концевых пробелов так, чтобы строки оказались посередине листа и их длина стала равна l. Вася считает, что строка находится посередине листа, если количество ведущих пробелов не превосходит количества концевых пробелов и, если при сдвиге строки на один символ вправо (т.е. при добавлении одного пробела в начало строки), указанное свойство нарушается.

Формат входного файла

Первая строка входного файла содержит два целых числа l и k ($1 \le l \le 100$, $1 \le k \le 1000$). Следующие k строк содержат текст Васиного письма. Каждая строка письма содержит хотя бы один символ, отличный от пробела. Длина каждой строки во входном файле не превосходит 100.

Формат выходного файла

Если Вася сможет написать письмо, удовлетворяющее всем его требованиям, то выведите отформатированный текст письма, иначе, выведите фразу “Impossible.” (без кавычек).

Примеры

letter.in	letter.out

`20 3`
`__Привет!__`
`_Напиши_мне.__`
`____Пока_=)_`
	`______Привет!_______`
	`____Напиши_мне._____`
	`______Пока_=)_______`

`5 1` `Привет.`	`Impossible.`

В приведённом примере для большей наглядности все пробелы заменены на символы подчёркивания!

Задача G. Железная дорога

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

При строительстве новой железной дороги возникли проблемы. Дорога пролегает по холмистой местности, однако сами пути должны идти строго горизонтально. Поэтому руководство строительной компании приняло решение выровнять поверхность земли на этом участке. Главная проблема состоит в том, что привозить или вывозить землю на стройку стоит 10000$ за кубический метр. Поскольку бюджет железной дороги невелик, этого нельзя себе позволить.

Поэтому главный инженер принял решение выровнять поверхность, используя только землю, из которой состоят холмы. Теперь самая сложная задача состоит в том, чтобы выяснить высоту над уровнем моря, на которой будет пролегать дорога. Это ответственное задание было поручено Вам.

Через каждый метр от начала участка была измерена высота над уровнем моря. Напишите программу, которая по данным измерений рассчитывает искомую высоту.

Формат входного файла

Первая строка входного файла содержит количество N ($1 \le N \le 30000$) точек, в которых была замерена высота. Вторая строка содержит результаты замеров — i-ое число строки содержит высоту над уровнем моря точки, находящейся на расстоянии $(i-1)$ метр от начала участка. Все высоты — целые неотрицательные числа, не превосходящие 10000. Считайте, что между соседними точками измерений земная поверхность строго прямолинейна.

Формат выходного файла

Выведите в выходной файл ответ на задачу с точностью не меньшей, чем 10^{-5}.

Примеры

railway.in	railway.out
4 0 1 1 0	0.6666666667

Задача H. Цифровое колдовство

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

В современном мире все становится цифровым. Шаманы в одной сказочной северной стране тоже стараются не отставать от жизни. Последние изыскания показали, что в качестве материальных компонентов для колдовства с большой эффективностью можно использовать длинные веревки, сплетенные из моржовых усов. Каждая веревка кодирует магическое целое число следующим образом: сначала она определенным образом делится на части, соответствующие цифрам магического числа, в каждой из которых можно завязать от нуля до девяти узелков (глобализация вынуждает шаманов использовать обычную десятичную систему счисления). Конечно, при этом для задания очень больших чисел могут понадобиться длинные веревки.

Можно надеяться, что вводная часть вам ясна. Шаман племени Ы наслал порчу на племя У. Шаману племени У удалось выяснить, какое магическое число n было использовано при проведении страшного ритуала. Теперь ему нужно создать веревку с таким числом k, чтобы $n + k$ делилось на фундаментальную магическую константу m. Необходимо сделать это как можно быстрее, поэтому он хочет выполнить задачу, завязав наименьшее число узелков, даже если для этого придется взять самую длинную веревку из потайного хранилища.

Поможете ли вы спасти племя У (а в след за ним, может быть, и весь мир) от ужасного шамана племени Ы?

Формат входного файла

Во входном файле записаны два целых числа — n и m ($0 \le n \le 1000, 1 < m \le 1000$).

Формат выходного файла

В выходной файл выведите одно целое неотрицательное число k — ответ на задачу.

Примеры

shaman.in	shaman.out
5 8	11