Задача A. Расписание турнира

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

Как известно, командные спортивные соревнования часто проводятся по круговой системе, когда любые две команды должны сыграть между собой ровно один матч. Круговой турнир проводится в несколько туров, в одном туре каждая команда может сыграть не более одного матча. Например, если в турнире участвуют 4 команды, то турнир можно провести в три тура: в первом туре команда 1 играет с командой 2, а команда 3 играет с командой 4, во втором туре 1 играет с 3, а 2 играет с 4, в третьем туре — 1 играет с 4, а 2 играет с 3.

Организаторам олимпиады Сочи-2014 необходимо организовать несколько командных турниров по круговой системе с участием различного числа команд. График олимпиады очень плотный, поэтому каждый турнир нужно провести в минимально возможное число туров. Для составления расписания каждого турнира они решили обратиться за помощью к программистам.

Формат входных данных

Во входном файле записано одно натуральное число N — количество команд, участвующих в турнире ($2 \le N \le 100$).

Формат выходных данных

В первой строке выведите минимальное количество туров K, необходимых для проведения кругового турнира из N команд. Каждая из K следующих строк содержит описание одного тура. В начале строки выведите количество игр n_i, которое необходимо сыграть в i-м туре. Далее идет n_i пар чисел — команды, которые играют в этом туре. Команды, играющие между собой, разделяются символом "-" (минус), а разные игры разделяются пробелом.

Примеры

tournament.in	tournament.out
4	3 2 1-2 3-4 2 1-3 2-4 2 1-4 2-3
3	3 1 1-2 1 2-3 1 3-1

Задача B. Округление

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

Заданы n чисел a_1, a_2, …, a_n.

Говорят, эти числа были получены параллелью Q+ и и описывают количество любителей различных языков программирования в ЛКШ.

К сожалению, завуч Андрей Сергеевич хочет видеть эти числа не в виде долей единицы и даже не в процентах, а в виде целых долей числа m, где число m — некоторое положительное целое.

Потому каждое из этих чисел делится на их сумму, а затем умножается на m. Кроме того, если число после умножения оказалось нецелым, то вы имеете право его округлить до целого в любую сторону. Однако завуч поставил вполне логичное условие — сумма получившихся чисел должна быть равна m.

Напишите программу, которая делает требуемое округление!

Формат входных данных

В первой строке входного файла записаны два целых числа n и m ($1 \le n \le 5000$, $1 \le m \le 10000$). В следующей строке записана последовательность n целых неотрицательных чисел a_1, …, a_n ($1 \le \sum a_i \le 100000$).

Формат выходных данных

В выходной файл необходимо вывести округленную последовательность — n целых чисел через пробел. Сумма получившихся чисел должна быть равна m.

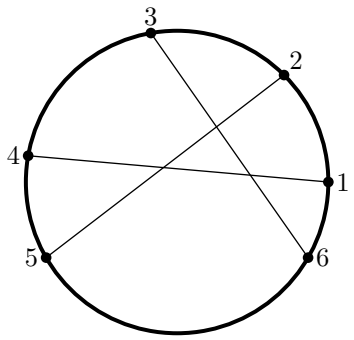
Пример

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

Задача C. Хорды

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

На окружности отмечено $2n$ различных точек, пронумерованных от 1 до $2n$ против часовой стрелки. Петя нарисовал n хорд, i-я из которых соединяет точки с номерами a_i и b_i. При этом каждая точка является концом ровно одной хорды.



Теперь Петя заинтересовался, сколько пар хорд пересекаются. Помогите ему определить это количество.

Формат входных данных

Первая строка входного файла содержит целое число n — количество проведенных хорд ($1 \le n \le 100\,000$). Следующие n строк содержат по два целых числа — a_i и b_i.

Формат выходных данных

Выведите одно число — количество пар хорд, которые пересекаются.

Примеры

chords.in	chords.out
3 1 4 2 5 3 6	3
2 1 2 3 4	0
2 1 4 2 3	0

Задача D. Следующая строка

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

Назовём строку из нулей и единиц *простой*, если она лексикографически меньше любого своего собственного суффикса. Например, строка «`00101`» простая, а «`00000`» — нет (любой её собственный суффикс меньше всей строки).

Необходимо по простой строке найти следующую в лексикографическом порядке простую строку такой же длины.

Формат входных данных

Входной файл содержит простую строку длины n ($2 \leqslant n \leqslant 10\,000$).

Формат выходных данных

В выходном файле должна находиться следующая в лексикографическом порядке простая строка длины n. Гарантируется, что она существует.

Пример

next.in	next.out
00111	01011

Задача E. Опять строка...

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

Вася в очередной раз занялся строками. В этот раз у него есть две строки, и он хочет проверить, можно ли вторую разрезать ровно на три части таким образом, чтобы из получившихся частей можно было склеить первую. Например, строку `beast` можно разрезать и преобразовать в `betas`, а `royalitem` в `romeitaly` — нельзя.

Формат входных данных

В двух строках входного файла даны две строки — первая и вторая. Каждая из них непуста и состоит не более чем из 5 000 маленьких латинских букв. Длины строк одинаковы, и каждая буква встречается одинаковое количество раз.

Формат выходных данных

Выведите `YES`, если требуемое разрезание возможно, и `NO` в противном случае. В случае `YES` на последующих трех строках выведите части, на которые нужно разрезать вторую строку, в порядке, в котором их нужно склеить для получения первой. Части не могут быть пустыми строками. Если возможных разрезаний несколько, разрешается выводить любое.

Пример

stragain.in	stragain.out
beast betas	YES be as t
royalitem romeitaly	NO

Задача F. Операция Xor

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

Пусть есть два неотрицательных целых числа a и b. Представим их в двоичной системе счисления $a = (a_1 a_2 \dots a_l)_2$ и $b = (b_1 b_2 \dots a_l)_2$ $(a_i, b_i \in \{0, 1\})$. Без ограничения общности можно полагать, что они имеют одинаковую длину, так как к меньшему по длине можно подписать слева необходимое количество нулей.

Результат операции $c = a \oplus b$ — число, i-тая цифра двоичной записи которого равна $(a_i + b_i) \mod 2$. В программировании такая операция называется операцией **xor**.

Ваша задача найти набор из n чисел a_i, удовлетворяющих условиям:

1. $a_i \neq a_j$ ни для каких $i \neq j, 1 \leq i, j \leq n$
2. $0 < a_i \leq 2^{31} - 1$
3. $a_1 \oplus a_2 \oplus \dots \oplus a_n = A$

Формат входных данных

Во входном файле содержаться два натуральных числа n $(1 \leq n \leq 100000)$ и A $(0 \leq A \leq 2^{31} - 1)$.

Формат выходных данных

В выходной файл через пробел запишите n чисел, удовлетворяющих вышеописанному условию или «-1», если такого набора не существует.

Пример

xor.in	xor.out
1 5	5
7 1	12 71 35 3 54 138 214