Задача A. Следующее сочетание

Имя входного файла:	`nextcomb.in`
Имя выходного файла:	`nextcomb.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайта

Дано множество целых чисел от 1 до N. Рассмотрим подмножество этого множества, состоящее из K элементов, в возрастающем порядке.

Выведите следующее в лексикографическом порядке подмножество из K элементов.

Формат входного файла

В первой строке входного файла содержатся целые положительные числа N и K ($1 \leqslant K \leqslant N \leqslant 50$). Во второй строке содержится K целых чисел от 1 до N в возрастающем порядке — подмножество из K элементов.

Формат выходного файла

Выведите следующее в лексикографическом порядке после данного подмножество из K элементов. Если следующего подмножества нет, выведите 0.

Пример

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

Задача B. Предыдущая перестановка

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

Найдите предыдущую в лексикографическом порядке перестановку. Перестановка вида $N, N-1, ..., 3, 2, 1$ является предыдущей для $1, 2, 3, ..., N-1, N$.

Формат входного файла

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 10^5$) количество элементов в перестановке. Во второй строке записана перестановка.

Формат выходного файла

В выходной файл вывести N чисел — искомую перестановку.

Пример

prev.in	prev.out
3 1 2 3	3 2 1

Задача C. Перестановка по номеру

Имя входного файла:	`bynumber.in`
Имя выходного файла:	`bynumber.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайта

Вывести перестановку по номеру.

Формат входного файла

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 12$) — количество элементов в перестановке. Во второй строке — число K ($0 \leqslant K \leqslant N! - 1$) — номер перестановки в нумерации с нуля.

Формат выходного файла

В выходной файл выведите, разделяя пробелами, N чисел искомой перестановки.

Пример

bynumber.in	bynumber.out
3 0	1 2 3

Задача D. Монетки

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

В Волшебной стране используются монетки достоинством $A_1, A_2, \ldots, A_M$. Волшебный человечек пришёл в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму N. Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

Формат входного файла

Сначала вводится число $N (1 \leqslant N \leqslant 10^9)$, затем — число $M (1 \leqslant M \leqslant 10)$ и далее M попарно различных чисел $A_1, A_2, \ldots, A_M$ $(1 \leqslant A_i \leqslant 10^9)$.

Формат выходного файла

Выведите сначала K — количество монет, которое придётся отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите K чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них. Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число –1 (минус один).

Примеры

coins.in	coins.out
5 2 1 2	3 2 2 1
7 2 1 2	-1
5 2 3 4	0

Задача E. Разбиения на слагаемые

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

Перечислите все разбиения целого положительного числа N на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

- слагаемые в разбиениях идут в невозрастающем порядке;
- разбиения перечисляются в лексикографическом порядке.

Формат входного файла

Во входном файле находится единственное число N $(1 \leqslant N \leqslant 40)$.

Формат выходного файла

В выходной файл выведите искомые разбиения по одному на строку.

Пример

partition.in	partition.out
4	1 1 1 1 2 1 1 2 2 3 1 4