Задача A. Получить перестановку по её номеру

Имя входного файла:	`by-number.in`
Имя выходного файла:	`by-number.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дана последовательность натуральных чисел от 1 до N. Требуется найти её K-ю в лексикографическом порядке перестановку.

Формат входного файла

В первой строке входных данных содержится число N ($1 \leqslant N \leqslant 12$) — количество элементов в перестановке, во второй — число K ($1 \leqslant K \leqslant N!$) — номер перестановки.

Формат выходного файла

Выведите N чисел — искомую перестановку.

Примеры

by-number.in	by-number.out
3 4	2 3 1

Задача B. Номер размещения

Имя входного файла:	`by-arrangement.in`
Имя выходного файла:	`by-arrangement.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дано размещение первых N натуральных чисел по K. Требуется найти его номер в лексикографическом порядке.

Формат входного файла

В первой строке входных данных находятся целые числа N и K ($1 \leqslant K \leqslant N \leqslant 12$). Во второй строке записаны K целых чисел из диапазона от 1 до N — размещение.

Формат выходного файла

Выведите единственное число — номер данного размещения.

Примеры

by-arrangement.in	by-arrangement.out
3 2 3 2	6

Задача C. Следующая перестановка

Имя входного файла:	`next.in`
Имя выходного файла:	`next.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дана перестановка из первых N натуральных чисел. Найдите по ней следующую в лексикографическом порядке.

Будем считать, что за перестановкой $(N, N-1, ..., 3, 2, 1)$ следует тождественная перестановка, то есть, $(1, 2, 3, ..., N)$.

Формат входного файла

В первой строке входных данных содержится целое число N ($1 \leqslant N \leqslant 10\,000$). Во второй строке находится перестановка (последовательность натуральных чисел от 1 до N, разделенных пробелами).

Формат выходного файла

Требуется вывести искомую перестановку.

Примеры

next.in	next.out
3 1 3 2	2 1 3

Задача D. Скобки

Имя входного файла:	`parentheses.in`
Имя выходного файла:	`parentheses.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Правильной скобочной последовательностью называется такая последовательность скобок, которая могла бы встречаться в каком-нибудь арифметическом выражении. Например, последовательности `()()` и `(())` являются правильными скобочными последовательностями, а последовательности `((` и `)()(` — нет.

Говорят, что одна скобочная последовательность *лексикографически меньше другой*, если существует такое число k, что первые k скобок в них совпадают, а $k + 1$-я скобка в первой меньше, чем во второй. Считается, что открывающая скобка меньше закрывающей.

Программист Вася выписал на бумажке все правильные скобочные последовательности длины $2n$ таким образом, что каждая написанная скобочная последовательность лексикографически меньше следующей за ней. Он очень обрадовался, что теперь может по лексикографическому номеру скобочной последовательности найти саму последовательность (последовательности, записанные на бумажке, нумеруются, начиная с 1). Однако бумажка потерялась... А повторять эту огромную работу Васе лень... Помогите бедному Васе написать программу, которая бы выдавала правильную скобочную последовательность по ее лексикографическому номеру!

Формат входного файла

Во входном файле задано число n ($1 \leqslant n \leqslant 20$) и произвольное натуральное число A — лексикографический номер искомой последовательности. Гарантируется, что последовательность с этим номером существует.

Формат выходного файла

В первой строке выходного файла должна содержаться искомая скобочная последовательность.

Примеры

parentheses.in	parentheses.out
2 1	(())
2 2	()()

Задача E. Гладкие числа

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

Назовем число гладким, если его цифры, начиная со старшего разряда, образуют неубывающую последовательность. Упорядочим все такие числа в возрастающем порядке и присвоим каждому номер. Вам требуется по номеру N вывести N-ое гладкое число.

Формат входного файла

На вход программы поступает номер $N (1 \leq N \leq 2147483647)$.

Формат выходного файла

Выведите соответствующее номеру N гладкое число.

Примеры

smooth.in	smooth.out
3	3
11	12

Задача F. ПСП 3

Имя входного файла:	`parens3.in`
Имя выходного файла:	`parens3.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Определим по индукции множество $\mathcal{T}$ *правильных скобочных последовательностей из трёх типов скобок*:

- $\varepsilon \in \mathcal{T}$ (пустая строка)
- $A \in \mathcal{T} \Rightarrow (A) \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow [A] \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow \{A\} \in \mathcal{T}$
- $A \in \mathcal{T}, B \in \mathcal{T} \Rightarrow AB \in \mathcal{T}$

Пусть теперь $\mathcal{T}_n$ — это множество правильных скобочных последовательностей из $2n$ символов — n открывающих и n закрывающих скобок.

Упорядочим элементы множества $\mathcal{T}_n$ лексикографически с некоторым порядком символов '`(`', '`)`', '`[`', '`]`', '`{`' и '`}`'.

По данным числам n и p, а также порядку, заданному на скобках, найдите p-ый в этом порядке элемент множества $\mathcal{T}_n$.

Формат входного файла

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leqslant n \leqslant 20$, $0 \leqslant p \leqslant 9 \cdot 10^{18}$). Скобочные последовательности нумеруются с нуля.

Во второй строке записаны шесть символов — '`(`', '`)`', '`[`', '`]`', '`{`' и '`}`' — в некотором порядке. Их порядок задаёт лексикографический порядок на множестве $\mathcal{T}_n$.

Формат выходного файла

В первой строке выходного файла выведите $2n$ символов без пробелов — p-ю правильную скобочную последовательность длины $2n$ из трёх типов скобок.

Если для данного n не существует p-я правильная скобочная последовательность, выведите в первой строке "`N/A`".

Примеры

parens3.in	parens3.out
1 0 ()}[{]	()
2 16 ()[]{}	{}[]