Задача A. Ровно k единиц

Имя входного файла:	`comb1.in`
Имя выходного файла:	`comb1.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

По данным натуральным n и k выведите все двоичные последовательности длины n, содержащие ровно k единиц в лексикографическом порядке.

Формат входных данных

Входной файл содержит два числа, n и k ($1 \leqslant n \leqslant 100, 0 \leqslant k \leqslant n$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом. Гарантируется, что количество чисел в выходном файле не превосходит 200000

Примеры

comb1.in	comb1.out
4 2	0 0 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0

Задача B. Сочетания-2

Имя входного файла:	`comb2.in`
Имя выходного файла:	`comb2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

По данным натуральным k и n ($1 \leqslant k \leqslant n$) выведите все **убывающие** последовательности длины k состоящие из чисел $1 \ldots n$ в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — k и n ($1 \leqslant k \leqslant n \leqslant 1000$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом. Гарантируется, что количество чисел в выходном файле не превосходит 500 000.

Примеры

comb2.in	comb2.out
3 5	3 2 1 4 2 1 4 3 1 4 3 2 5 2 1 5 3 1 5 3 2 5 4 1 5 4 2 5 4 3

Задача C. Без двух единиц подряд

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

По данному натуральному числу n выведите все двоичные последовательности длины n, не содержащие двух единиц подряд, в лексикографическом порядке.

Формат входных данных

Одно натуральное число n ($n \leqslant 20$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Примеры

fibseq.in	fibseq.out
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

Задача D. Правильные скобочные последовательности

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	64 мегабайта

Дано натуральное число n. Выведите каждую k-ю правильную скобочную последовательность, состоящую из n открывающих круглых скобок и n закрывающих скобок в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — n и k ($1 \leqslant n \leqslant 15$, $1 \leqslant k \leqslant 100$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки.

Примеры

стандартный ввод	стандартный вывод
3 1	((()))
	(()())
	(())()
	()(())
	()()()
3 2	(()())
	()(())

Задача E. Следующее разбиение на слагаемые

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

Формат входных данных

Входной файл содержит одну строку — разбиение числа n на слагаемые ($1 \leqslant n \leqslant 100\,000$). Слагаемые в разбиении следуют в неубывающем порядке.

Формат выходных данных

Выведите в выходной файл одну строку — разбиение числа n на слагаемые, следующее в лексикографическом порядке после приведенного во входном файле. Если во входном файле приведено последнее разбиение n на слагаемые, выведите `No solution`.

Примеры

nextpartition.in	nextpartition.out
5=1+1+3	5=1+2+2
5=5	No solution

Задача F. Следующая перестановка

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

Найдите следующую перестановку. Лексикографически первая перестановка является следующей для обратной.

Формат входных данных

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 100\,000$) — количество элементов в перестановке. Во второй строке записана перестановка из N чисел.

Формат выходных данных

В выходной файл вывести N чисел — искомую перестановку.

Примеры

nextperm.in	nextperm.out
3 3 2 1	1 2 3
2 1 2	2 1

Задача G. Предыдущее сочетание

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

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

Выведите предыдущее в лексикографическом порядке подмножество из K элементов.

Формат входных данных

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 выходных данных

Выведите предыдущее в лексикографическом порядке перед данным подмножество из K элементов. Если предыдущего подмножества нет, выведите 0.

Примеры

prevcomb.in	prevcomb.out
6 4 1 4 5 6	1 3 5 6

Задача H. Перестановка по номеру

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Выведите перестановку по её номеру.

Формат входных данных

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

Формат выходных данных

В выходной файл выведите N чисел через пробел — искомую перестановку.

Примеры

stdin	stdout
3 0	1 2 3

Задача I.

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

Определим по индукции множество $\mathcal{R}$ *правильных скобочных последовательностей*:

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

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

Упорядочим элементы множества $\mathcal{R}_n$ лексикографически с порядком символов ‘$($’ $<$ ‘$)$’.

По данным числам n и p найдите p-ый в этом порядке элемент множества $\mathcal{R}_n$.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leqslant n \leqslant 20$, $0 \leqslant p \leqslant 2 \cdot 10^9$). Скобочные последовательности нумеруются с нуля.

Формат выходных данных

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

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

Примеры

parens.in	parens.out
3 0	((()))
3 1	(()())