

Задача А. Перестановки

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

Во входном файле задано число n ($1 \leq n \leq 8$). Выведите в выходной файл в лексикографическом порядке все перестановки чисел от 1 до n .

Пример

perm.in	perm.out
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Задача В. Обратная перестановка

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

Обратной перестановкой к перестановке $p_1, p_2, \dots, p_n$ называется перестановка $q_1, q_2, \dots, q_n$, в которой q_i — это номер места, на котором стоит число i в перестановке p . К примеру, обратной к перестановке 1 4 2 3 является перестановка 1 3 4 2.

По данной перестановке длины n найдите обратную к ней перестановку.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 10$). Во второй строке заданы n целых чисел $p_1, p_2, \dots, p_n$ через пробел. Гарантируется, что эти числа образуют перестановку чисел 1, 2, $\dots$, n .

Формат выходных данных

В первой строке выходного файла выведите n чисел через пробел — перестановку, обратную данной.

Примеры

inverse.in	inverse.out
1 1	1
2 1 2	1 2
4 1 4 2 3	1 3 4 2

Задача С. Следующая перестановка

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

По данной перестановке длины n выведите перестановку, следующую за ней в лексикографическом порядке. Если перестановка — последняя в лексикографическом порядке, выведите лексикографически минимальную перестановку такой же длины.

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 10$). Во второй строке заданы n целых чисел $p_1, p_2, \dots, p_n$ через пробел. Гарантируется, что эти числа образуют перестановку чисел 1, 2, $\dots$, n .

Формат выходных данных

В первой строке выходного файла выведите n чисел через пробел — искомую перестановку.

Примеры

success.in	success.out
3 1 2 3	1 3 2
3 3 2 1	1 2 3

Задача D. Сочетания

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

Во входном файле заданы числа n и k . Выведите в выходной файл все сочетания по k из чисел от 1 до n в лексикографическом порядке. $1 \leq k \leq n \leq 16$.

Пример

<code>choose.in</code>	<code>choose.out</code>
4 2	1 2 1 3 1 4 2 3 2 4 3 4

Задача E. Поколение комбинаторов

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

Сочетанием из n элементов по k называется убывающая последовательность из k чисел из диапазона от 1 до n .

Сгенерируйте все сочетания из n элементов по k в антилексикографическом порядке, т.е. так, что для любых двух выведенных сочетаний первые l чисел равны, а $l + 1$ -е в предыдущем больше, чем в следующем.

Формат входных данных

Во входном файле содержатся два целых числа n и k . $1 \leq k \leq n \leq 15$.

Формат выходных данных

В выходной файл выведите все сочетания из n элементов по k в антилексикографическом порядке, по одному сочетанию на строку.

Пример

<code>generation.in</code>	<code>generation.out</code>
3 2	3 2 3 1 2 1

Задача F. Сочетание

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

Сочетанием из n элементов по k назовём возрастающую последовательность из k целых чисел от 1 до n , включительно. Упорядочим все сочетания из n по k лексикографически как векторы из k чисел.

По заданным числам n , k и p выведите p -е в лексикографическом порядке сочетание из k натуральных чисел от 1 до n .

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа n , k и p ($1 \leq n \leq 32$, $0 \leq k \leq n$). Гарантируется, что p -е сочетание из n по k существует. Сочетания нумеруются с нуля.

Формат выходных данных

В первой строке выходного файла выведите k чисел через пробел — элементы p -го сочетания из n по k в порядке возрастания.

Примеры

<code>comb.in</code>	<code>comb.out</code>
4 2 0	1 2
4 2 1	1 3
4 2 2	1 4
4 2 3	2 3
4 2 4	2 4
4 2 5	3 4