

### Задача А. 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  $N$  ( $1 \leq N \leq 100\,000$ ) — количество элементов в перестановке. Во второй строке записана перестановка из  $N$  чисел.

#### Формат выходных данных

В выходной файл вывести  $N$  чисел — искомую перестановку.

#### Примеры

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

### Задача В. Предыдущая перестановка

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

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

#### Формат входных данных

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

#### Формат выходных данных

В выходной файл вывести  $N$  чисел — искомую перестановку.

#### Примеры

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

### Задача С. Следующее сочетание

Имя входного файла: `nextcomb.in`  
Имя выходного файла: `nextcomb.out`  
Ограничение по времени: 1 секунда  
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 следующее в лексикографическом порядке подмножество из  $K$  элементов.

#### Формат входных данных

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

#### Формат выходных данных

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

#### Примеры

<code>nextcomb.in</code>	<code>nextcomb.out</code>
6 4 1 4 5 6	2 3 4 5
6 2 5 6	0

### Задача D. Конфеты Кирилла

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

У Кирилла было  $k$  конфет, и он захотел их раздать ученикам своей параллели. Однако заметил, что конфет у него меньше чем учеников в параллели. Кирилл сел на скамейку и задумался. Просидев полчаса и доев последнюю конфету он подумал — интересно, а сколько было способов раздать все  $k$  конфет  $n$  ученикам параллели С, если конфеты нельзя делить, а каждому школьнику можно дать не более одной конфеты.

#### Формат входных данных

В единственной строке записаны числа  $n, k$  ( $1 \leq k \leq n \leq 64$ ).

#### Формат выходных данных

Выведите единственное число — ответ на задачу.

#### Примеры

<code>combination.in</code>	<code>combination.out</code>
5 3	10

### Задача Е. Предыдущая правильная скобочная последовательность

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

По правильной скобочной последовательности вывести последовательность, предшествующую ей в лексикографическом порядке.

#### Формат входных данных

На вход подается одна строка, состоящая из  $2N$  ( $N \leq 10000$ ) символов - исходная правильная скобочная последовательность.

#### Формат выходных данных

Вывести в единственной строке искомую последовательность. Если исходная последовательность - первая в лексикографическом порядке, то следует вывести последнюю в лексикографическом порядке правильную скобочную последовательность такой же длины.

#### Примеры

<code>prev.in</code>	<code>prev.out</code>
<code>()()</code>	<code>((()))</code>
<code>()(())</code>	<code>((()))()</code>

#### Примеры

	<code>brackets.in</code>	<code>brackets.out</code>
3		<code>((()))</code> <code>((()())</code> <code>(())(())</code> <code>()(())</code> <code>()()()</code>

### Задача F. Генерация правильных скобочных последовательностей

Имя входного файла: `brackets.in`  
Имя выходного файла: `brackets.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 64 мегабайта

Выведите все правильные скобочные последовательности длины  $2n$  в лексикографическом порядке. Используются только круглые скобки. Считается, что открывающая скобка лексикографически меньше закрывающей.

#### Формат входных данных

Единственное число  $n$  ( $1 \leq n \leq 10$ ).

#### Формат выходных данных

Выведите скобочные последовательности в лексикографическом порядке, по одной в строке.