

## Задача А. 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 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. Перестановка по номеру

Имя входного файла: `bynumber.in`  
Имя выходного файла: `bynumber.out`  
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  $N$  ( $1 \leq N \leq 12$ ) — количество элементов в перестановке. Во второй строке записано число  $K$  ( $0 \leq K < N!$ ) — номер перестановки в нумерации с нуля.

### Формат выходных данных

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

### Примеры

<code>bynumber.in</code>	<code>bynumber.out</code>
3 0	1 2 3

## Задача D. 30 кресел

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

Потерпев фиаско в погоне за стульями Остап не пал духом, а ввязался в новую авантюру. Услышав, что неподалёку на аукционе распродают  $n$  старинных кресел, не долго думая он решил попытать удачу и проверить: не скрываются ли сокровища в одном из них. Придя на торги, Остап понял, что денег у него хватит на выкуп ровно  $k$  из  $n$  кресел. Своим самым счастливым числом Остап считает число  $m$ , поэтому он снова обращается к Вам за помощью и просит выбрать  $m$ -е сочетание  $k$  из  $n$  кресел.

### Формат входных данных

Во входном файле заданы числа  $n$ ,  $k$  и  $m$ .  $1 \leq k \leq n \leq 30$ ,  $0 \leq m \leq \binom{n}{k} - 1$ .

### Формат выходных данных

Выведите в выходной файл в возрастающем порядке номера кресел, входящие в  $m$ -е в лексикографическом порядке сочетание по  $k$  из чисел от 1 до  $n$ . Сочетания занумерованы, начиная с 0.

### Примеры

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

## Задача Е. Разбиения на слагаемые

Имя входного файла: `partition.in`  
Имя выходного файла: `partition.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 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 входных данных

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

### Формат выходных данных

В выходной файл выведите искомые разбиения по одному на строку.

### Примеры

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

### Замечание

На питоне, чтобы задача проходила по времени, надо использовать метод `write` вместо функции `print`.