

Задача А. Предыдущий и следующий двоичный вектор

Имя входного файла: `nextvector.in`

Имя выходного файла: `nextvector.out`

Во входном файле задан двоичный вектор. Выведите в выходной файл предыдущий и следующий двоичный вектор в лексикографическом порядке. Если какого-либо из них не существует, выведите вместо него «-». Длина вектора во входном файле — от 1 до 200000.

Пример

<code>nextvector.in</code>	<code>nextvector.out</code>
10001	10000 10010
0	- 1

Задача В. Предыдущая и следующая перестановки

Имя входного файла: `nextperm.in`
Имя выходного файла: `nextperm.out`

Во входном файле задано число n и затем перестановка чисел от 1 до n . Выведите в выходной файл предыдущую и следующую перестановку чисел от 1 до n . Если какой либо из них не существует, выведите вместо нее n нулей. $1 \leq n \leq 100\,000$.

Пример

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

Задача С. Следующее сочетание

Имя входного файла: `nextchoose.in`
Имя выходного файла: `nextchoose.out`

Во входном файле заданы числа n , k и затем сочетание, состоящее из k чисел от 1 до n .
($1 \leq k \leq n \leq 10000$)

Выведите в выходной файл следующее сочетание в лексикографическом порядке из n чисел по k .

Если его не существует, выведите -1.

Пример

<code>nextchoose.in</code>	<code>nextchoose.out</code>
4 2 2 3	2 4
4 2 3 4	-1

Задача D. Следующее разбиение на множества

Имя входного файла: `nextsetpartition.in`
Имя выходного файла: `nextsetpartition.out`

Рассмотрим множество первых n натуральных чисел: $N_n = \{1, 2, \dots, n\}$. Разбиением на множества называется представление этого множества, как объединения одного или более, попарно непересекающихся подмножеств множеств. Например для $n = 5$ существуют следующие разбиения:

$$\begin{aligned}\{1, 2, 3, 4, 5\} &= \{1, 2, 3\} \cup \{4, 5\} \\ \{1, 2, 3, 4, 5\} &= \{1, 3, 5\} \cup \{2, 4\} \\ \{1, 2, 3, 4, 5\} &= \{1, 2, 3, 4, 5\} \\ \{1, 2, 3, 4, 5\} &= \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}\end{aligned}$$

Всего существует 52 разбиения множества N_5 . Заметьте, что мы не различаем разбиения на множества, которые отличаются только порядком подмножеств.

Упорядочим все разбиения на множества N_n лексикографически. Для этого во-первых в каждом разбиении упорядочим множества лексикографически. Будем говорить, что подмножество $A \subset N_n$ лексикографически меньше подмножества $B \subset N_n$, если верно одно из следующих условий:

- существует i такое, что $i \in A$, $i \notin B$, для всех $j < i$: $j \in A$ если и только если $j \in B$, и существует $k > i$ такое что $k \in B$;
- $A \subset B$ и $i < j$ для всех $i \in A$ и $j \in B \setminus A$.

Разбиения упорядочены лексикографически следующим образом. Разбиение $N_n = A_1 \cup A_2 \cup \dots \cup A_k$ лексикографически меньше разбиения $N_n = B_1 \cup B_2 \cup \dots \cup B_l$ если существует такое i , что $A_1 = B_1$, $A_2 = B_2$, \dots , $A_{i-1} = B_{i-1}$ и $A_i < B_i$.

Дано разбиение N_n , ваша задача найти следующее разбиение на множества в лексикографическом порядке.

Формат входного файла

Во входном файле содержится несколько тестов. Каждый тест в первой строчке содержит n и k — количество чисел в разбиваемом множестве, и количество подмножеств в разбиении. ($1 \leq n \leq 200$). Следующие k строк содержат элементы разбиения. Элементы в каждом подмножестве упорядочены по возрастанию.

Тесты разделены пустой строкой. Последняя строка содержит два нуля.

Сумма всех n по всем тестам не превосходит 2000.

Формат выходного файла

Для каждого теста выведите в выходной файл следующее разбиение. Если разбиение во входном файле является последним в лексикографическом порядке, то выведите первое в лексикографическом порядке разбиение. Используйте такой же формат, как и во входном файле. Разделяйте ответы для разных тестов пустой строкой.

Примеры

nextsetpartition.in	nextsetpartition.out
5 2 1 2 3 4 5	5 2 1 2 3 4 5
5 2 1 3 5 2 4	5 4 1 4 2 3 5
5 1 1 2 3 4 5	5 2 1 2 3 5 4
5 5 1 2 3 4 5	5 4 1 2 3 4 5
0 0	

Задача Е. Следующая правильная скобочная последовательность

Имя входного файла: `nextbrackets.in`
Имя выходного файла: `nextbrackets.out`

Во входном файле задана правильная скобочная последовательность. Выведите в выходной следующую за ней в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, «(» < «)». Если такой нет, выведите «-». Количество открывающихся скобок в последовательности — от 1 до 100 000.

Пример

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

Задача F. Следующая мультиперестановка

Имя входного файла: `nextmultiperm.in`

Имя выходного файла: `nextmultiperm.out`

Во входном файле задано число n и затем мультиперестановка, составленная из чисел от 1 до n . Выведите в выходной файл следующую в лексикографическом порядке мультиперестановку того же мультимножества. Если искомой перестановки не существует, выведите n нулей. $1 \leq n \leq 100\,000$.

Пример

<code>nextmultiperm.in</code>	<code>nextmultiperm.out</code>
6 1 3 2 1 3 2	1 3 2 2 1 3

Задача G. Следующее разбиение на слагаемые

Имя входного файла: `nextpartition.in`
Имя выходного файла: `nextpartition.out`

Разбиения числа n на слагаемые — это набор целых положительных чисел, сумма которых равна n . При этом разбиения, отличающиеся лишь порядком слагаемых, считаются одинаковыми, поэтому можно считать, что слагаемые в разбиении упорядочены по неубыванию.

Например, существует 7 разбиений числа 5 на слагаемые:

$5 = 1 + 1 + 1 + 1 + 1$
 $5 = 1 + 1 + 1 + 2$
 $5 = 1 + 1 + 3$
 $5 = 1 + 2 + 2$
 $5 = 1 + 4$
 $5 = 2 + 3$
 $5 = 5$

В приведенном примере разбиения упорядочены *лексикографически* — сначала по первому слагаемому в разбиении, затем по второму, и так далее. В этой задаче вам потребуется по заданному разбиению на слагаемые найти следующее в лексикографическом порядке разбиение.

Формат входного файла

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

Формат выходного файла

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 числа n на слагаемые, выведите «No solution».

Примеры

<code>nextpartition.in</code>	<code>nextpartition.out</code>
<code>5=1+1+3</code>	<code>5=1+2+2</code>
<code>5=5</code>	<code>No solution</code>