

Задача G. Разбиения на множества (2 балла)

Имя входного файла: `part2sets.in`
Имя выходного файла: `part2sets.out`

Во входном файле заданы числа n и k . Выведите в выходной файл все разбиения n -элементного множества на k неупорядоченных множеств. Разбиения можно выводить в любом порядке. Внутри разбиения множества можно выводить в любом порядке. Внутри множества числа надо выводить в возрастающем порядке. Следуйте формату из примера. $1 \leq k \leq n \leq 10$.

Пример

part2sets.in	part2sets.out
4 2	1 2 3 4 2 1 3 4 3 1 2 4 4 1 2 3 1 2 3 4 1 3 2 4 2 3 1 4

Задача H. Перестановка по номеру (1 балл)

Имя входного файла: `num2perm.in`
Имя выходного файла: `num2perm.out`

Во входном файле задано числа n и k . Выведите в выходной файл k -ю в лексикографическом порядке перестановку чисел от 1 до n . Перестановки занумерованы от 0 до $n! - 1$. $1 \leq n \leq 18$, $0 \leq k \leq n! - 1$.

Пример

num2perm.in	num2perm.out
3 4	3 1 2

Задача I. Номер по перестановке (1 балл)

Имя входного файла: `perm2num.in`
Имя выходного файла: `perm2num.out`

Во входном файле задано число n и затем перестановка чисел от 1 до n . Выведите в выходной файл номер заданной перестановки в лексикографическом порядке всех перестановок чисел от 1 до n . Перестановки занумерованы, начиная с 0. $1 \leq n \leq 18$.

Пример

perm2num.in	perm2num.out
3 1 3 2	1

Задача J. Сочетание по номеру (1 балл)

Имя входного файла: `num2choose.in`
Имя выходного файла: `num2choose.out`

Во входном файле заданы числа n , k и m . Выведите в выходной файл m -е в лексикографическом порядке сочетание по k из чисел от 1 до n . Сочетания занумерованы, начиная с 0. $1 \leq k \leq n \leq 30$, $0 \leq m \leq \binom{n}{k} - 1$.

Пример

num2choose.in	num2choose.out
4 2 3	2 3

Задача K. Номер по сочетанию (1 балл)

Имя входного файла: `choose2num.in`
Имя выходного файла: `choose2num.out`

Во входном файле заданы числа n , k и затем сочетание, состоящее из k чисел от 1 до n . Выведите в выходной файл номер этого сочетания в лексикографическом порядке всех сочетаний из n чисел по k ($1 \leq k \leq n \leq 30$). Сочетания нумеруются, начиная с 0.

Пример

choose2num.in	choose2num.out
4 2 2 3	3

Задача L. Правильная скобочная последовательность по номеру (1 балл)

Имя входного файла: `num2brackets.in`
Имя выходного файла: `num2brackets.out`

Во входном файле заданы числа n и k . Выведите в выходной файл k -ю в лексикографическом порядке правильную скобочную последовательность среди всех правильных скобочных последовательностей с n открывающимися скобками, упорядоченных в лексикографическом порядке, «(» < «)». Последовательности занумерованы, начиная с 0. $1 \leq n \leq 20$. Искомая последовательность существует.

Пример

num2brackets.in	num2brackets.out
4 3	((()))()

Задача M. Номер по правильной скобочной последовательности (1 балл)

Имя входного файла: `brackets2num.in`
Имя выходного файла: `brackets2num.out`

Во входном файле задана правильная скобочная последовательность. Выведите в выходной ее номер в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, «(» < «)». Последовательности занумерованы, начиная с 0. Количество открывающихся скобок в последовательности — от 1 до 20.

Пример

brackets2num.in	brackets2num.out
((()))()	3

Задача N. Правильная скобочная последовательность с двумя типами скобок по номеру (1 балл)

Имя входного файла: num2brackets2.in
Имя выходного файла: num2brackets2.out

Во входном файле заданы числа n и k . Выведите в выходной файл k -ю в лексикографическом порядке правильную скобочную последовательность среди всех правильных скобочных последовательностей с двумя типами скобок с n открывающимися скобками, упорядоченных в лексикографическом порядке, «(» < «)» < «[» < «]». Последовательности занумерованы, начиная с 0. $1 \leq n \leq 20$. Искомая последовательность существует.

Пример

num2brackets2.in	num2brackets2.out
4 100	([])([])

Задача O. Номер по правильной скобочной последовательности с двумя типами скобок (1 балл)

Имя входного файла: brackets2num2.in
Имя выходного файла: brackets2num2.out

Во входном файле задана правильная скобочная последовательность с двумя типами скобок. Выведите в выходной ее номер в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, «(» < «)» < «[» < «]». Последовательности занумерованы, начиная с 0. Количество открывающихся скобок в последовательности — от 1 до 20.

Пример

brackets2num2.in	brackets2num2.out
([])([])	100

Задача P. Разбиение на слагаемые по номеру (1 балл)

Имя входного файла: num2part.in
Имя выходного файла: num2part.out

Рассмотрим все разбиения числа n на слагаемые, в каждом разбиении упорядочим их в порядке не убывания. Будем считать, что разбиение $a_1 + a_2 + \dots + a_n$ лексикографически меньше $b_1 + b_2 + \dots + b_m$, если для некоторого $k \forall j \leq k : a_j = b_j$ и либо $k = n$, либо $a_{k+1} < b_{k+1}$. Во входном файле заданы числа n и r . $1 \leq n \leq 100$, разбиение с номером r — существует. Выведите r -ое разбиение числа n на слагаемые, разбиения нумеруются с 0.

Пример

num2part.in	num2part.out
4 3	2+2

Задача Q. Номер по разбиению на слагаемые (1 балл)

Имя входного файла: part2num.in
Имя выходного файла: part2num.out

Рассмотрим все разбиения числа n на слагаемые, в каждом разбиении упорядочим их в порядке не убывания. Будем считать, что разбиение $a_1 + a_2 + \dots + a_n$ лексикографически меньше $b_1 + b_2 + \dots + b_m$, если для некоторого $k \forall j \leq k : a_j = b_j$ и либо $k = n$, либо $a_{k+1} < b_{k+1}$. Во входном файле задано разбиение на слагаемые. Выведите номер этого разбиения, среди всех разбиений упорядоченных лексикографически. Разбиения нумеруются с 0. Гарантируется, что в разбиении слагаемые упорядочены в порядке не убывания, и $1 \leq n \leq 100$.

Пример

part2num.in	part2num.out
2+2	3

Задача R. 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

nextvector.in	nextvector.out
10001	10000 10010
0	- 1

Задача S. 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

nextperm.in	nextperm.out
4 1 3 2 4	1 2 4 3 1 3 4 2
2 1 2	0 0 2 1

Задача Т. 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

nextchoose.in	nextchoose.out
4 2 2 3	2 4
4 2 3 4	-1

Задача У. 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 0 0	5 4 1 2 3 4 5

Задача V. 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

nextbrackets.in	nextbrackets.out
(())()	()((()))

Задача W. 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

nextmultiperm.in	nextmultiperm.out
6 1 3 2 1 3 2	1 3 2 2 1 3

Задача X. 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>