Задача 1. Двоичные коды

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

Во входном файле задано число n. Выведите в выходной файл в лексикографическом порядке все двоичные вектора длины n. $1 \leq n \leq 16$.

Пример

allvectors.in	allvectors.out
3	000 001 010 011 100 101 110 111

Задача 2. Коды Грея для двоичных векторов

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

Во входном файле задано число n. Выведите в выходной файл в порядке произвольного кода Грея все двоичные вектора длины n. $1 \leq n \leq 16$.

Пример

gray.in	gray.out
3	000 001 011 010 110 111 101 100

Задача 3. Коды Антигрея

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

Во входном файле задано число n. Выведите в выходной файл все троичные вектора длины n, так чтобы в соседних отличались значения на всех n позициях. $1 \leq n \leq 10$.

Пример

antigray.in	antigray.out
2	11 22 00 12 20 01 10 21 02

Задача 4. Цепной код

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

Во входном файле задано число n. Выведите в выходной файл все двоичные вектора длины n, в порядке какого-нибудь цепного кода. $1 \leq n \leq 15$.

Пример

chaincode.in	chaincode.out
3	000 001 010 101 011 111 110 100

Задача 5. Телеметрия

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

Во входном файле заданы числа n и k. Выведите в выходной файл все k-ичные вектора длины n, так чтобы у двух подряд идущих векторов, значения на всех кроме одной позиции совпадали, а значения на оставшейся позиции отличались ровно на 1 ($n \geq 2$, $2 \leq k \leq 9$, $1 \leq k^n \leq 100000$).

Пример

telemetry.in	telemetry.out
3 3	000 100 200 210 110 010 020 120 220 221 121 021 011 111 211 201 101 001 002 102 202 212 112 012 022 122 222

Задача 6. Двоичные вектора

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

Во входном файле задано число n ($1 \leq n \leq 16$). В первой строке выходного файла выведите количество двоичных векторов длины n в которых нет двух единиц подряд. В следующих строках выведите сами эти вектора в лексикографическом порядке по одному в строке.

Пример

vectors.in	vectors.out
3	5 000 001 010 100 101

Задача 7. Перестановки

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

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

permutations.in	permutations.out
3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

Задача 8. Сочетания

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

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

choose.in	choose.out
4 2	1 2 1 3 1 4 2 3 2 4 3 4

Задача 9. Правильные скобочные последовательности

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

Во входном файле задано число n. Выведите в выходной файл все правильные скобочные последовательности с n открывающимися скобками в лексикографическом порядке, «(» < «)». $1 \leq n \leq 10$.

Пример

brackets.in	brackets.out
4	((((()))) (((()()))) (((())())) (((()))()) ((()(()))) ((()()())) ((()())()) ((())(())) ((())()()) ()((())) ()(()()) ()((())() ()()(()) ()()()()

Задача 10. Разбиения на слагаемые

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

Во входном файле задано число n ($2 \leq n \leq 40$). Выведите в выходной файл все разбиения числа n на слагаемые по одному в строке. Слагаемые следует выводить в возрастающем порядке. Разбиения отличающиеся только порядком слагаемых считаются одинаковыми.

Пример

partition.in	partition.out
4	1+1+1+1 1+1+2 1+3 2+2 4

Задача 11. Подмножества

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

Во входном файле задано число n. Выведите в выходной файл все подмножества множества $\{1, 2, \ldots, n\}$ в лексикографическом порядке. $1 \leq n \leq 10$.

Пример

subsets.in	subsets.out
3	 1 1 2 1 2 3 1 3 2 2 3 3

Задача 12. Разбиения на множества

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. $1 \le k \le n \le 10$.

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 13. Перестановка по номеру

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 14. Номер по перестановке

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 15. Сочетание по номеру

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 16. Номер по сочетанию

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 ($1 \le k \le n \le 30$). Сочетания нумеруются, начиная с 0.

Пример

choose2num.in	choose2num.out
4 2 2 3	3

Задача 17. Правильная скобочная последовательность по номеру

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 18. Номер по правильной скобочной последовательности

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 19. Правильная скобочная последовательность с двумя типами скобок по номеру

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 20. Номер по правильной скобочной последовательности с двумя типами скобок

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 21. Разбиение на слагаемые по номеру

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 $a_1+a_2+...+a_n$ лексикографически меньше $b_1+b_2+...+b_m$, 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 22. Номер по разбиению на слагаемые

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 $a_1+a_2+...+a_n$ лексикографически меньше $b_1+b_2+...+b_m$, 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 23. 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 24. 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 25. 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 26. Следующее разбиение на множества

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

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

$$\{1,2,3,4,5\} = \{1,2,3\} \cup \{4,5\}$$
$$\{1,2,3,4,5\} = \{1,3,5\} \cup \{2,4\}$$
$$\{1,2,3,4,5\} = \{1,2,3,4,5\}$$
$$\{1,2,3,4,5\} = \{1\} \cup \{2\} \cup \{3\} \cup \{4\} \cup \{5\}$$

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 $N_n = A_1 \cup A_2 \cup \ldots \cup A_k$ лексикографически меньше разбиения $N_n = B_1 \cup B_2 \cup \ldots \cup B_l$ если существует такое i, что $A_1 = B_1$, $A_2 = B_2$, $\ldots$, $A_{i-1} = B_{i-1}$ 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. ($1 \le n \le 200$). 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 3 5 2 4 5 1 1 2 3 4 5 5 5 1 2 3 4 5 0 0	5 2 1 2 3 4 5 5 4 1 4 2 3 5 5 2 1 2 3 5 4 5 4 1 2 3 4 5

Задача 27. 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 28. 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 29. 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 ($1 \le n \le 100\,000$). 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

nextpartition.in	nextpartition.out
5=1+1+3	5=1+2+2
5=5	No solution