Задача A. Без двух единиц подряд

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

По данному натуральному числу n выведите все двоичные последовательности длины n, не содержащие двух единиц подряд, в лексикографическом порядке.

Формат входных данных

Одно натуральное число n ($n \leqslant 20$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом.

Пример

стандартный ввод	стандартный вывод
3	0 0 0 0 0 1 0 1 0 1 0 0 1 0 1

Задача B. Сочетания-2

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

По данным натуральным k и n ($1 \leqslant k \leqslant n$) выведите все **убывающие** последовательности длины k состоящие из чисел $1 \dots n$ в лексикографическом порядке.

Формат входных данных

Во входном файле два числа — k и n ($1 \leqslant k \leqslant n \leqslant 1000$).

Формат выходных данных

Каждая последовательность должна выводиться в отдельной строке, вывод должен завершаться символом новой строки. Числа, входящие в последовательность, должны быть разделены одним пробелом. Гарантируется, что количество чисел в выходном файле не превосходит 500 000.

Пример

стандартный ввод	стандартный вывод
3 5	3 2 1 4 2 1 4 3 1 4 3 2 5 2 1 5 3 1 5 3 2 5 4 1 5 4 2 5 4 3

Задача C. Перестановка по номеру

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Выведите перестановку по её номеру.

Формат входных данных

В первой строке входного файла записано число N ($1 \leqslant N \leqslant 12$) — количество элементов в перестановке. Во второй строке записано число K ($0 \leqslant K < N!$) — номер перестановки в нумерации с нуля.

Формат выходных данных

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

Пример

стандартный ввод	стандартный вывод
3 0	1 2 3

Задача D. Генерация правильных скобочных последовательностей - 2

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

По данному числу n выведите все правильные скобочные последовательности из круглых и квадратных скобок длины n. Придерживайтесь следующего порядка скобок: "([])"(см. тест из условия)

Формат входных данных

Одно целое число n ($0 \leqslant n \leqslant 16$).

Формат выходных данных

Выведите все правильные скобочные последовательности из круглых и квадратных скобок длины n в лексикографическом порядке. Каждая послеждовательность должна выводиться в новой строке.

Пример

стандартный ввод	стандартный вывод
4	(()) ([]) ()() ()[] [()] [[]] []() [][]

Задача E. Номер правильной последовательности

Имя входного файла:	`brackets2num2.in`
Имя выходного файла:	`brackets2num2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам дана правильная скобочная последовательность с двумя типами скобок. Выведите её 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.

Формат входных данных

Во входном файле записана строка из круглых и квадратных скобок. Длина строки не превосходит 20 символов.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

brackets2num2.in	brackets2num2.out
`()`	0
`[]`	1
`()()`	1
`()[]`	2
`([])()[]`	100
`[[[(()[]([]))]]]`	266079

Задача F. Монетки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2.2 секунд
Ограничение по памяти:	256 мегабайт

64 мегабайта

В Волшебной стране используются монетки достоинством $A_1, A_2, \ldots, A_M$. Волшебный человечек пришел в магазин и обнаружил, что у него есть ровно по две монетки каждого достоинства. Ему нужно заплатить сумму N. Напишите программу, определяющую, сможет ли он расплатиться без сдачи.

Формат входных данных

Сначала вводится целое число N ($1 \leqslant N \leqslant 10^9$), затем — целое число M ($1 \leqslant M \leqslant 10$) и далее M попарно различных целых чисел $A_1, A_2, \ldots, A_M$ ($1 \leqslant A_i \leqslant 10^9$).

Формат выходных данных

Выведите сначала K — количество монет, которое придется отдать Волшебному человечку, если он сможет заплатить указанную сумму без сдачи. Далее выведите K чисел, задающих достоинства монет. Если решений несколько, выведите вариант, в котором Волшебный человек отдаст наименьшее возможное количество монет. Если таких вариантов несколько, выведите любой из них.

Если без сдачи не обойтись, то выведите одно число 0. Если же у Волшебного человечка не хватит денег, чтобы заплатить указанную сумму, выведите одно число –1 (минус один).

Примеры

стандартный ввод	стандартный вывод
5 2 1 2	3 1 2 2
7 2 1 2	-1
5 2 3 4	0

Задача G. Новогодняя гирлянда

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

256 мегабайт

Дети в детском саду как-то раз решили повесить к Новому году гирлянду. Но это оказалось для них очень трудной задачей. На помощь пришёл Дед Мороз, который теперь каждый Новый год приносит с собой гирлянду и помогает её повесить.

Гирлянда представляет собой ломаную в плоскости, состоящую из n звеньев. Гирлянда начинается в точке $(0, 0)$, возле электророзетки и должна заканчиваться в точке $(n, 0)$. Число n называется длиной гирлянды. Каждое звено может располагаться либо горизонтально, либо под углом 45° к оси OX. Длина горизонтальной проекции любого звена равна 1. При этом не должно быть вершины ломаной с отрицательной координатой y, а также двух последовательных вершин с нулевой координатой y. Поднимающимся (опускающимся) назовём звено ломаной, у которого координата у правого конца больше (соответственно, меньше) координаты у левого конца. Звено, у которого координаты у концов совпадают, назовём горизонтальным. Обозначим поднимающееся звено буквой `u`, опускающееся — буквой `d`, а горизонтальное — буквой `h`. Тогда гирлянда кодируется строкой из n символов. У Деда Мороза есть волшебная книга, в которой перечислены все гирлянды длины n в виде строк. Хотя книга и волшебная, строки в ней располагаются в обычном лексикографическом порядке, по возрастанию. Дед Мороз отметил на полях книги галочкой гирлянду, которую повесил в прошлый раз. В этот Новый год он хочет повесить следующую в книге гирлянду. Найдите эту гирлянду без использования волшебной книги.

Формат входных данных

В первой строке вводится целое число n ($2 \leqslant n \leqslant 100\,000$). Во второй — строчка из n букв (все буквы: `u`, `d`, либо `h`) — прошлогодняя гирлянда.

Формат выходных данных

Выведите в виде строки гирлянду, которую Дед Мороз Павлович должен прихватить с собой в этот Новый год, либо `No solution`, если такой гирлянды не существует.

Пример

стандартный ввод	стандартный вывод
6 uhduhd	uhhdud

Задача H. Гроб гроб кладбище treap

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан массив целых чисел. Необходимо реализовать структуру данных, в которой требуется за $O(\log n)$ выполнять запросы:

1. сумма на подотрезке $[L, R]$ (в задаче принята 0-индексация);
2. вставить элемент x в позицию pos (т.е. в результате вставки, элемент x должен оказаться pos-м);
3. удалить элемент x, находящийся на позиции i;
4. присвоить элемент x на подотрезке $[L, R]$;
5. прибавить число x на подотрезке $[L, R]$;
6. $next\_permutation$ на подотрезке $[L, R]$;
7. $prev\_permutation$ на подотрезке $[L, R]$.

$next\_permutation$ и $prev\_permutation$ должны работать так же, как одноименные STL-алгоритмы; В частности, $next\_permutation([4, 3, 2, 1])$ есть $[1, 2, 3, 4]$, а не $[4, 3, 2, 1]$; Аналогично, $prev\_permutation([1, 2, 2, 2, 3, 3, 4]) = [4, 3, 3, 2, 2, 2, 1]$.

Формат входных данных

В первой строке записано число $n (1 \leqslant n \leqslant 3 \cdot 10^4)$ — количество элементов в массиве. Во второй строке записано n чисел, не превосходящих по модулю $3 \cdot 10^4$ — исходные значения элементов массива.

В третьей строке записано число $q (1 \leqslant q \leqslant 10^5)$ - количество запросов. В последующих строках записаны сами запросы, по одному на каждой строке.

Запросы задаются в следующем формате:

- 1 L R ($0 \leqslant l \leqslant r < arraySize$ - найти сумму всех чисел массива на отрезке [l, r];
- 2 x pos ($|x| \leqslant 3 \cdot 10^4, 0 \leqslant pos \leqslant arraySize$): вставить элемент x на позицию pos;
- 3 pos ($0 \leqslant pos < arraySize$): удалить элемент, находящийся на позиции pos;
- 4 x L R ($|x| \leqslant 3 \cdot 10^4, 0 \leqslant l \leqslant r < arraySize$): присвоить элементам на подотрезке $[L, R]$ значение x;
- 5 x L R ($|x| \leqslant 3 \cdot 10^4, 0 \leqslant l \leqslant r < arraySize$): прибавить к элементам на подотрезке $[L, R]$ число x;
- 6 L R: выполнить $next\_permutation$ на подотрезке $[L, R]$;
- 7 L R: выполнить $prev\_permutation$ на подотрезке $[L, R]$.

В приведенном описании $arraySize$ — размер массива на момент запроса. Иными словами вам гарантируется, что все запросы корректные.

Формат выходных данных

Для каждого запроса типа 1 выведите соответствующую сумму в отдельной строке. По выполнении всех запросов, выведите итоговые значения элементов массива также в отдельной строке.

Пример

стандартный ввод	стандартный вывод
7 1 2 3 4 5 6 7 8 4 5 1 3 2 3 3 5 2 0 4 7 0 6 6 0 3 3 2 1 1 5 1 0 6	28 40 5 3 7 6 7 5 7