Задача A. Перестановки и числа на местах

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Задано целые положительные числа n и k. Сколько различных перестановок длины n, в которых ровно k чисел стоят на своих местах?

Формально: i-е число в перестановке $p_1, p_2, \ldots, p_n$ стоит на своём месте, если $p_i = i$.

Формат входных данных

В первой строке записаны целые числа n и k ($0 \leqslant k \leqslant n \leqslant 20$).

Формат выходных данных

Выведите одно целое число: количество перестановок длины n, в которых ровно k чисел стоят на своих местах.

Примеры

стандартный ввод	***стандартный вывод***	***пояснение***
3 0	2	2 3 1 3 1 2
3 1	3	1 3 2 2 1 3 3 2 1
3 2	0	
3 3	1	1 2 3

Задача B. Рекорды и циклы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

В этой задаче нужно построить биекцию между перестановками длины n с k рекордами и перестановками длины n с k циклами.

Перестановка p длины n — это последовательность из n целых чисел $p(1)$, $p(2)$, …, $p(n)$ такая, что каждое из чисел 1, 2, …, n встречается в ней ровно один раз.

Цикл в перестановке p — это последовательность из целых чисел c_1, c_2, …, c_m такая, что $p(c_1) = c_2$, $p(c_2) = c_3$, …, $p(c_m) = c_1$. Особый случай — цикл единичной длины: $p(i) = i$. Можно показать, что любую перестановку можно представить как набор попарно непересекающихся циклов.

Рекорд в перестановке p — это её элемент p_j такой, что он больше всех предшествующих ему элементов перестановки: $p_j > p_i$ для любого $i < j$. Заметим, что p_1 по определению всегда считается рекордом.

Можно доказать, что для любых заданных n и k количество перестановок длины n с ровно k циклами равно количеству перестановок длины n с ровно k рекордами. Поэтому существует биекция между этими двумя множествами. Напомним, что биекция — это взаимно однозначное отображение между двумя множествами: каждому элементу первого множества сопоставлен ровно один элемент второго множества, а каждому элементу второго — ровно один элемент первого. Придумайте и реализуйте такое отображение.

Формат входных данных

Первая строка ввода содержит целое число t — количество перестановок ($1 \leqslant t \leqslant 300\,000$).

Каждая из t последующих строк описывает одну перестановку. Описание начинается с числа n — длины перестановки. Далее следует символ «`r`» (рекорды) или «`c`» (циклы), обозначающий тип перестановки. После этого задана сама перестановка — последовательность из n целых чисел, в которой каждое целое число от 1 до n встречается ровно один раз.

Соседние числа или символы в строке разделяются одиночными пробелами. Общая длина всех перестановок во вводе не превосходит $300\,000$.

Формат выходных данных

Формат вывода совпадает с форматом ввода.

Первая строка вывода должна содержать целое число t — количество перестановок ($1 \leqslant t \leqslant 300\,000$).

Далее для каждой из t заданных перестановок выведите соответствующую ей перестановку на отдельной строке. Если тип перестановки во вводе обозначен символом «`r`», найдите количество рекордов в ней и выведите соответствующую перестановку типа «`c`» с таким же количеством циклов. Если же тип перестановки во вводе обозначен символом «`c`», найдите количество циклов в ней и выведите соответствующую перестановку типа «`r`» с таким же количеством рекордов.

Каждая из t выведенных строк с перестановками должна начинаться с числа n — длины перестановки. Далее должен следовать символ «`r`» или «`c`», обозначающий тип перестановки. После этого должна быть выведена сама перестановка. Разделяйте соседние числа или символы в строке одиночными пробелами.

Выведенное соответствие должно быть согласованным: если перестановка p типа «`r`» соответствует перестановке q типа «`c`», обратное соответствие также должно иметь место, никакая другая перестановка типа «`c`» не может соответствовать p и никакая другая перестановка типа «`r`» не может соответствовать q. Отметим, однако, что соответствие не обязательно должно быть согласованным между ответами на различные тесты.

Пример

стандартный ввод	*стандартный вывод*	альтернативный ответ
6 3 r 1 2 3 3 c 1 2 3 3 r 2 1 3 3 r 2 3 1 3 c 2 1 3 2 c 2 1	6 3 c 1 2 3 3 r 1 2 3 3 c 2 1 3 3 c 3 2 1 3 r 2 1 3 2 r 2 1	6 3 c 1 2 3 3 r 1 2 3 3 c 3 2 1 3 c 1 3 2 3 r 1 3 2 2 r 2 1

Пояснение к примеру

В этом примере первые пять перестановок имеют длину $n = 3$. Первая перестановка имеет тип «`r`» и содержит три рекорда. Вторая перестановка имеет тип «`c`» и содержит три цикла. Эти две перестановки соответствуют друг другу. Третья и четвёртая перестановки имеют тип «`r`» и содержат по два рекорда каждая. Пятая перестановка имеет тип «`c`» и содержит два цикла. Она соответствует третьей перестановке. Четвёртая перестановка соответствует перестановке 3, 2, 1 типа «`c`» — ещё одной перестановке с двумя циклами.

Последняя перестановка имеет длину $n = 2$, тип «`c`» и содержит один цикл. Она соответствует перестановке 2, 1 типа «`r`» с одним рекордом.

Заметим, что это не единственная возможная биекция: один из альтернативных ответов приведён выше. В нём третьей перестановке во вводе (это 2, 1, 3 типа «`r`») мы сопоставляем 3, 2, 1 типа «`c`». После этого четвёртой перестановке (2, 3, 1 типа «`r`») также следует сопоставить другую перестановку, и мы выбираем для этого перестановку 1, 3, 2 типа «`c`». Наконец, пятой перестановке (2, 1, 3 типа «`c`») можно сопоставить лишь 1, 3, 2 типа «`r`», поскольку две другие перестановки с двумя рекордами уже сопоставлены иным перестановкам с двумя циклами.

Задача C. Перестановка по номеру

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановкой из n элементов называется некоторый набор чисел от 1 до n, записанных в произвольном порядке, в котором каждое число встречается ровно один раз.

Говорят, что одна перестановка *лексикографически меньше* другой, если первые k чисел в них совпадают, а $(k+1)$-е число в первой меньше, чем во второй.

Набор перестановок называется *лексикографически упорядоченным*, если в нём каждая перестановка лексикографически меньше, чем следующая.

Ясно, что из всех n-элементных перестановок можно единственным образом построить лексикографически упорядоченный набор. Например, для $n = 3$ таким набором будет ((1 2 3), (1 3 2), (2 1 3), (2 3 1), (3 1 2), (3 2 1)).

Лексикографическим номером перестановки из n элементов называется ее номер в лексикографически упорядоченном наборе всех n-элементных перестановок. Например, перестановка (2 3 1) будет иметь лексикографический номер 4.

Вам требуется написать программу, которая будет находить перестановку по числу n ($1 \leqslant n \leqslant 20$) и её лексикографическому номеру k.

Формат входных данных

В первой строке заданы два числа: n и k. Число k не превосходит 10^{18} по модулю.

Формат выходных данных

Вам необходимо выдать саму перестановку — n чисел, или сообщение «`INVALID number.`» (без кавычек), если перестановки с таким номером не существует.

Примеры

стандартный ввод	**стандартный вывод**
3 1	1 2 3
3 2	1 3 2
3 4	2 3 1
3 7	INVALID number.

Задача D. Номер по перестановке

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Вам требуется написать программу, которая будет находить по перестановке её лексикографический номер среди всех перестановок из n различных чисел от 1 до n.

Формат входных данных

В первой строке задано число n ($1 \leqslant n \leqslant 20$). Во второй строке заданы n чисел, образующих перестановку.

Формат выходных данных

Вам необходимо выдать одно число — лексикографический номер перестановки (нумерация начинается с единицы).

Примеры

стандартный ввод	**стандартный вывод**
3 1 2 3	1
3 1 3 2	2
3 2 3 1	4

Задача E. Перестановки и подлестницы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановка длины n — это последовательность из n целых чисел, в которой каждое число от 1 до n встречается ровно один раз. Как известно, существует ровно $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$ перестановок длины n.

Назовём *подлестницей* последовательность из n целых чисел: первое число лежит в диапазоне от 0 до $n-1$, второе — в диапазоне от 0 до $n-2$, и так далее. В частности, предпоследнее число может быть либо нулём, либо единицей, а последнее число всегда равно нулю. Заметим, что существует ровно $n! = n \cdot (n-1) \cdot (n-2) \cdots 2 \cdot 1$ подлестниц — столько же, сколько перестановок длины n.

Придумайте, как сопоставить каждой перестановке подлестницу, чтобы различным перестановкам соответствовали различные подлестницы. Напишите программу, которая строит по любой перестановке подлестницу, а потом по этой подлестнице восстанавливает исходную перестановку.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза.

При первом запуске решение по перестановке строит подлестницу. В первой строке записано слово «`encode`». Вторая строка содержит целое число n — длину перестановки ($1 \leqslant n \leqslant 200\,000$). Третья строка содержит перестановку — n целых чисел от 1 до n, каждое ровно один раз. Решение должно вывести подлестницу — строку из n целых чисел, первое из которых лежит в диапазоне от 0 до $n-1$, второе — от 0 до $n-2$, и так далее, в частности, последнее число просто равно нулю.

При втором запуске решение восстанавливает перестановку по подлестнице. В первой строке записано слово «`decode`». Вторая строка содержит то же самое целое число n — длину перестановки ($1 \leqslant n \leqslant 200\,000$). Третья строка содержит подлестницу — n целых чисел, ровно те, которые решение вывело при первом запуске. Решение должно вывести исходную перестановку — строку из n целых чисел от 1 до n.

Пример

В каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Далее показаны два запуска одного и того же решения на первом тесте.

стандартный ввод	*стандартный вывод*

`encode`
`5`
`2 1 5 3 4`
	`4 1 0 1 0`

`decode`
`5`
`4 1 0 1 0`
	`2 1 5 3 4`

Обратите внимание: при первом запуске вы можете вывести не только ответ из примера, но и любую другую подлестницу, например, `0 0 0 0 0` или `4 3 2 1 0`. При втором запуске ваше решение получит именно ту подлестницу, которую оно выведет при первом запуске, и правильным считается только один ответ — исходная перестановка `2 1 5 3 4`.

Задача F. Ограниченные перестановки 1

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановкой длины n называется последовательность $p_1, p_2, \ldots, p_n$, в которой каждое целое число от 1 до n встречается ровно один раз.

Даны числа n и k. Посчитайте количество перестановок длины n, в которых каждый элемент отличается от своего номера не больше чем на k. Другими словами, для всех i от 1 до n верно $|p_i - i| \leqslant k$.

Формат входных данных

В первой строке заданы два целых числа n и k — размер перестановки и ограничение ($1 \leqslant n \leqslant 10$, $0 \leqslant k \leqslant 3$).

Формат выходных данных

Выведите одно целое число: требуемое количество перестановок.

Пример

стандартный ввод	*стандартный вывод*	*пояснение*
3 1	3	1 2 3 1 3 2 2 1 3
4 0	1	1 2 3 4

Задача G. Ограниченные перестановки 2

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановкой длины n называется последовательность $p_1, p_2, \ldots, p_n$, в которой каждое целое число от 1 до n встречается ровно один раз.

Даны числа n и k. Посчитайте количество перестановок длины n, в которых каждый элемент отличается от своего номера не больше чем на k. Другими словами, для всех i от 1 до n верно $|p_i - i| \leqslant k$.

Так как ответ может быть большим, выведите остаток от деления искомого количества на $998\,244\,353$.

Формат входных данных

В первой строке заданы два целых числа n и k — размер перестановки и ограничение ($1 \leqslant n \leqslant 100$, $0 \leqslant k \leqslant 9$).

Формат выходных данных

Выведите одно целое число: требуемое количество перестановок по модулю $998\,244\,353$.

Пример

стандартный ввод	*стандартный вывод*	*пояснение*
3 1	3	1 2 3 1 3 2 2 1 3
4 0	1	1 2 3 4

Задача H. Ограниченные перестановки 3

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановкой длины n называется последовательность $p_1, p_2, \ldots, p_n$, в которой каждое целое число от 1 до n встречается ровно один раз.

Даны числа n и k. Посчитайте количество перестановок длины n, в которых каждый элемент отличается от своего номера не больше чем на k. Другими словами, для всех i от 1 до n верно $|p_i - i| \leqslant k$.

Так как ответ может быть очень большим, выведите остаток от деления искомого количества на $998\,244\,353$.

Формат входных данных

В первой строке заданы два целых числа n и k — размер перестановки и ограничение ($1 \leqslant n \leqslant 10^9$, $0 \leqslant k \leqslant 3$).

Формат выходных данных

Выведите одно целое число: требуемое количество перестановок по модулю $998\,244\,353$.

Пример

стандартный ввод	*стандартный вывод*	*пояснение*
3 1	3	1 2 3 1 3 2 2 1 3
4 0	1	1 2 3 4

Задача I. Перестановки и различные порядки

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Рассмотрим множество всех перестановок из n чисел от 1 до n. *Тождественной* называется перестановка, в которой на первом месте стоит единица, на втором — двойка, и так далее.

Перестановки можно применять друг к другу. При применении перестановки (a_i) к перестановке (b_i) получается перестановка $a(b)$, в которой на i-е место становится элемент a_{b_i}. Это также называют *умножением* перестановок.

В частности, перестановки можно *возводить в степень*. Возвести перестановку a в степень n — значит, выполнить $x = a$, а потом $n - 1$ раз $x = a(x)$.

Известно, что для каждой перестановки существует такое число k, что если её возвести в степень k, получается тождественная перестановка. Минимальное положительное такое k называется *порядком* этой перестановки.

Необходимо сосчитать, сколько различных значений порядков получится для всех перестановок из n элементов. Например, для $n = 2$ есть перестановка порядка 1 $(1, 2)$ и перестановка порядка 2 $(2, 1)$. Для $n = 3$ бывают ещё перестановки порядка 3 (например, $(2, 3, 1)$).

Формат входных данных

Целое n ($1 \leqslant n \leqslant 1000$).

Формат выходных данных

Количество различных возможных порядков.

Примеры

стандартный ввод	*стандартный вывод*
2	2
3	3
4	4
10	16

Задача J. Максимальный порядок

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Перестановка длины n — это последовательность из n чисел, в которой каждое целое число от 1 до n встречается ровно один раз. Например, существует шесть перестановок длины 3: `1 2 3`, `1 3 2`, `2 1 3`, `2 3 1`, `3 1 2` и `3 2 1`.

Тождественная перестановка длины n — это последовательность `1 2 3 ... n`. Тождественная перестановка обозначается буквой e; её длина обычно понятна из контекста.

Произведением перестановок $p = p_1\ p_2\ \ldots\ p_n$ и $q = q_1\ q_2\ \ldots\ q_n$ называется перестановка $r = r_1\ r_2\ \ldots\ r_n$, в которой элементы получаются по формуле $r_i = p_{q_i}$. Произведение обозначается так: $p \cdot q$. Например, произведением перестановок $p =$ `2 4 1 3 5` и $q =$ `4 2 1 5 3` является перестановка $p \cdot q =$ `3 4 2 5 1`. Заметим, что порядок сомножителей важен: например, $q \cdot p =$ `2 5 4 1 3`.

Перестановки также можно возводить в степень. *Степень перестановки* определяется так: $p^0 = e$, а для любого целого $k > 0$ верно $p^k = p \cdot p^{k-1}$. Например, для перестановки $p =$ `2 1 4 5 3` первые несколько степеней выглядят так:

```
0   1 2 3 4 5
1   2 1 4 5 3
2   1 2 5 3 4
3   2 1 3 4 5
4   1 2 4 5 3
5   2 1 5 3 4
6   1 2 3 4 5
7   2 1 4 5 3
        ...
```

Порядок перестановки p — это минимальное целое $k > 0$ такое, что $p^k = e$. Например, порядок перестановки `1 2 3` равен единице, а порядок перестановки `2 1 4 5 3` равен шести.

По данной длине n найдите перестановку, которая имеет максимальный порядок среди всех перестановок длины n.

Формат входных данных

В первой строке задано целое число n — требуемая длина перестановки ($1 \leqslant n \leqslant 100$).

Формат выходных данных

В первой строке выведите максимальный порядок перестановки длины n. Во второй строке выведите любую перестановку длины n, имеющую такой порядок. Соседние числа в перестановке следует разделять пробелами.

Примеры

стандартный ввод	**стандартный вывод**
2	2 2 1
5	6 2 1 4 5 3