

Задача А. Красивые зеркала

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

У Creatnx есть n зеркал, пронумерованных от 1 до n . Каждый день Creatnx спрашивает ровно одно зеркало «Красивый ли я?». i -е зеркало скажет Creatnx, что он красивый с вероятностью $\frac{p_i}{100}$ для всех $1 \leq i \leq n$.

Creatnx спрашивает зеркала одно за другим, начиная с 1-о зеркала. Каждый день, если он спрашивает i -е зеркало, есть две возможности:

- i -е зеркало скажет Creatnx, что он красивый. В этом случае, если $i = n$ Creatnx остановится и станет счастливым, иначе он продолжит спрашивать $i + 1$ -е зеркало на следующий день;
- В другом случае Creatnx очень расстроится. На следующий день, Creatnx начнет спрашивать 1-е зеркало заново.

Вам нужно посчитать математическое ожидание количества дней, до того как Creatnx станет счастливым.

Это число нужно найти по модулю 998244353. Формально, пусть $M = 998244353$. Можно показать, что ответ может быть представлен в виде несократимой дроби $\frac{p}{q}$, где p и q целые числа и $q \not\equiv 0 \pmod{M}$. Выведите целое число, равное $p \cdot q^{-1} \pmod{M}$. Другими словами, выведите такое целое число x , что $0 \leq x < M$ и $x \cdot q \equiv p \pmod{M}$.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество зеркал. Во второй строке находится n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq 100$).

Формат выходных данных

Выведите ответ по модулю 998244353 в единственной строке.

Примеры

стандартный ввод	стандартный вывод
1 50	2
3 10 20 50	112

Замечание

В первом тесте, есть единственное зеркало и оно говорит, что Creatnx красивый с вероятностью $\frac{1}{2}$. Поэтому математическое ожидание количества дней, пока Creatnx не станет счастливым равно 2.

Задача В. Гладкие числа

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

Назовем число гладким, если его цифры, начиная со старшего разряда, образуют неубывающую последовательность. Упорядочим все такие числа в возрастающем порядке и присвоим каждому номер. Вам требуется по номеру N вывести N -ое гладкое число.

Формат входных данных

На вход программы поступает номер N ($1 \leq N \leq 2147483647$).

Формат выходных данных

Выведите соответствующее номеру N гладкое число.

Примеры

стандартный ввод	стандартный вывод
3	3
11	12

Задача С. Построение

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

Помогите Генриетте написать программу, генерирующую k -ю в лексикографическом порядке правильную скобочную последовательность, состоящую из $2n$ скобок.

Формат входных данных

В единственной строке через пробел записаны целые числа n и k , при этом $1 \leq n \leq 18$.

Формат выходных данных

Выведите искомую правильную скобочную последовательность. Гарантируется, что она существует.

Пример

стандартный ввод	стандартный вывод
3 4	()(())

Задача D. Эскалатор

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

Илья устал от олимпиадного программирования, ушёл из университета и устроился на работу в мажораналде метрополитен. Перед ним поставили задачу определения нагрузки на эскалатор.

Пусть n человек стоят в очереди на эскалатор. В каждую секунду происходит одно из двух: либо первый человек в очереди с вероятностью p заходит на эскалатор, либо первый человек в очереди с вероятностью $1 - p$ остаётся стоять на месте, не в силах совладать с боязнью эскалаторов, задерживая при этом всю очередь за ним.

i -й в очереди не сможет зайти на эскалатор, пока на него не зайдут люди с номерами от 1 до $i - 1$ включительно. За одну секунду может зайти только один человек. Так как эскалатор бесконечный, то, единожды зайдя на него, человек никогда с него не сойдёт, т. е. будет ехать на нем в эту и в любую последующую секунды. Илье нужно посчитать математическое ожидание количества людей, которые будут находиться на эскалаторе после t секунд.

Вам необходимо помочь ему в решении этой непростой задачи.

Формат входных данных

В единственной строке входных данных заданы три числа n, p, t ($1 \leq n, t \leq 2000, 0 \leq p \leq 1$). Числа n и t — целые, число p — вещественное, заданное ровно с двумя знаками после запятой.

Формат выходных данных

Выведите одно вещественное число — математическое ожидание количества людей, которые будут на эскалаторе через t секунд. Абсолютная или относительная погрешность не должна превышать 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
1 0.50 1	0.5000000000
1 0.50 4	0.9375000000
4 0.20 2	0.4000000000

Задача Е. Награды

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

На одном Очень Важном Предприятии решили наградить некоторых k его работников. Конечно же, решили сделать это в соответствии со следующей Очень Важной Процедурой.

Всех n работников выстроили в один ряд. Причем, получилось так, что каждый работник видит только своих непосредственных соседей в этом ряду. Для повышения уровня производства на Очень Важном Предприятии начальство решило сделать так, чтобы каждый награжденный считал, что наградили именно его и только его. Для этого необходимо, чтобы в ряду не было двух рядом стоящих награжденных работников.

Вам необходимо написать программу, которая будет считать количество способов раздать таким образом k наград среди n стоящих в ряд работников. Так как это число может быть весьма большим, необходимо найти его остаток от деления на простое число m .

Формат входных данных

В первой и единственной строке входного файла заданы три целых неотрицательных числа n , k и m — количество работников на Очень Важном Предприятии, количество наград и простой модуль ($1 \leq k \leq n \leq 100000$, $1 \leq m \leq 10^9$).

Формат выходных данных

В выходной файл выведите единственное целое число — ответ на задачу, взятый по модулю простого числа m .

Примеры

стандартный ввод	стандартный вывод
3 2 569	1
5 2 673	6

Задача F. Красивая скобочная последовательность

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

Определим правильную скобочную последовательность и ее глубину следующим образом:

- Пустая строка это правильная скобочная последовательность глубины 0;
- Если « s » это правильная скобочная последовательность глубины d тогда « (s) » это правильная скобочная последовательность глубины $d + 1$;
- Если « s » и « t » это две правильные скобочные последовательности тогда их конкатенация « st » это правильная скобочная последовательность с глубиной равной максимальной из глубин s и t .

Для (не обязательно правильной) скобочной последовательности s мы определяем ее глубину как максимальную глубину любой **правильной** скобочной последовательности, которая может быть получена с помощью удаления некоторых символов из s (возможно нуля). Например, скобочная последовательность $s = \langle ()() \rangle$ имеет глубину 2, потому что при удалении третьего символа мы получим правильную скобочную последовательность « $()()$ » глубины 2.

Дана строка a , состоящая из символов ‘(’, ‘)’ и ‘?’. Рассмотрим все (не обязательно правильные) скобочные последовательности, получающиеся заменой всех символов ‘?’ в строке a на ‘(’ или ‘)’. Посчитайте сумму глубин всех таких скобочных последовательностей. Так как это число может быть очень большим, найдите его по модулю 998244353.

Взломы по этой задаче могут быть сделаны, только если простая и сложная версия этой задачи сданы.

Формат входных данных

Единственная строка содержит непустую строку, состоящую только из символов ‘(’, ‘)’ и ‘?’. Длина строки не превосходит 10^6 .

Формат выходных данных

Выведите ответ по модулю 998244353 в единственной строке.

Примеры

стандартный ввод	стандартный вывод
??	1
(?(?))	9

Замечание

В первом тесте, мы можем получить 4 скобочные последовательности меняя все символы ‘?’ на ‘(’ или ‘)’:

- « $(($ ». Ее глубина 0;
- « $)$ »». Ее глубина 0;
- « $($ »». Ее глубина 0;
- « $()$ ». Ее глубина 1.

Поэтому, ответ равен $1 = 0 + 0 + 0 + 1$.

Во втором тесте, мы можем получить 4 скобочные последовательности меняя все символы ‘?’ на ‘(’ или ‘)’:

- «((()))». Ее глубина 2;
- «()()»). Ее глубина 2;
- «((()))». Ее глубина 3;
- «()()»). Ее глубина 2.

Поэтому, ответ равен $9 = 2 + 2 + 3 + 2$.

Задача G. Новогодние перестановки

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

Перестановка длины n — это массив n целых чисел, таких что каждое целое число от 1 до n появляется в нем ровно один раз.

Элемент y перестановки p достижим из элемента x , Если $x = y$, или $p_x = y$, или $p_{p_x} = y$ и так далее.

Определим **декомпозицию** перестановки p следующим образом: сначала у нас есть перестановка p , все элементы которой **не помечены**, и пустой список l . Затем мы делаем следующее: пока хотя бы один элемент **не помечен** в p , находим самый левый такой элемент, перечисляем все элементы, которые достижимы из него **в порядке их появления в p** , помечаем все эти элементы, затем циклически сдвигаем список этих элементов так, чтобы максимум появился в первой позиции, и добавляем этот список **как элемент** в l . После того, как все элементы помечены, l является результатом этой декомпозиции.

Например, если мы хотим построить декомпозицию $p = [5, 4, 2, 3, 1, 7, 8, 6]$, мы делаем следующее:

1. изначально $p = [5, 4, 2, 3, 1, 7, 8, 6]$ (жирным шрифтом выделены помеченные элементы), $l = []$;
2. самый левый не помеченный элемент — 5; 5 и 1 достижимы из него, поэтому список, который мы хотим сдвинуть, — $[5, 1]$; нет необходимости сдвигать его, так как максимум уже является первым элементом;
3. $p = [5, 4, 2, 3, 1, 7, 8, 6]$, $l = [[5, 1]]$;
4. самый левый не помеченный элемент — 4, список достижимых элементов из него — это $[4, 2, 3]$; максимум уже первый элемент, поэтому сдвигать его не нужно;
5. $p = [5, 4, 2, 3, 1, 7, 8, 6]$, $l = [[5, 1], [4, 2, 3]]$;
6. самый левый не помеченный элемент — 7, список достижимых элементов из него — это $[7, 8, 6]$; мы должны сдвинуть его, чтобы он стал $[8, 6, 7]$;
7. $p = [5, 4, 2, 3, 1, 7, 8, 6]$, $l = [[5, 1], [4, 2, 3], [8, 6, 7]]$;
8. все элементы помечены, так что $[[5, 1], [4, 2, 3], [8, 6, 7]]$ — результат декомпозиции.

Определим *новогоднее преобразование* перестановки следующим образом: построим декомпозицию этой перестановки; затем отсортируем все списки в декомпозиции по возрастанию первых элементов (мы не меняем местами элементы в этих списках, только сами списки); затем объединим списки в один список, который становится новой перестановкой. Например, *новогоднее преобразование* $p = [5, 4, 2, 3, 1, 7, 8, 6]$ строится следующим образом:

1. декомпозиция равна $[[5, 1], [4, 2, 3], [8, 6, 7]]$;
2. после сортировки, декомпозиция становится равна $[[4, 2, 3], [5, 1], [8, 6, 7]]$;
3. $[4, 2, 3, 5, 1, 8, 6, 7]$ — результат преобразования.

Назовем перестановку **хорошей**, если результат ее преобразования совпадает с самой перестановкой. Например, $[4, 3, 1, 2, 8, 5, 6, 7]$ это хорошая перестановка; а $[5, 4, 2, 3, 1, 7, 8, 6]$ плохая, так как результатом преобразования является $[4, 2, 3, 5, 1, 8, 6, 7]$.

Ваша задача состоит в следующем: при заданных n и k найти k -ю (лексикографически) хорошую перестановку длины n .

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Каждый набор входных данных содержит два целых числа n и k ($1 \leq n \leq 50$, $1 \leq k \leq 10^{18}$).

Формат выходных данных

Для каждого набора входных данных выведите ответ на него следующим образом: если число хороших перестановок длины n меньше k , выведите одно целое число -1 ; в противном случае выведите k -ю (в лексикографическом порядке) хорошую перестановку из n элементов.

Пример

стандартный ввод	стандартный вывод
5	2 1 3
3 3	3 1 2 5 4
5 15	-1
4 13	1 2 6 3 4 5
6 8	1 2 4 3
4 2	