

<https://codeforces.com/contest/1411/problem/A>

## А. Игровой чат

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Вам предстоит разработать фильтр для плохих сообщений в чате одной игры. Сообщение представляет из себя строку  $S$  длины  $n$ , состоящую из строчных английских букв и символов ')'. Строка считается *плохой*, если количество символов ')' в конце строки превышает количество остальных символов. Например, у строки «)bc)» три скобки в конце, три остальных символа, и эта строка не считается плохой.

### Входные данные

В первой строке дано целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 100$ ). Далее дано описание  $t$  тестовых случаев.

В первой строке дано число  $n$  ( $1 \leq n \leq 100$ ). Во второй строке дана строка  $S$  длины  $n$ , состоящая из строчных английских букв и символов ')

### Выходные данные

Для каждого из  $t$  тестовых случаев выведите «Yes», если строка является плохой, иначе выведите «No».

Вы можете выводить каждую букву в любом регистре (строчную или заглавную).

### Пример

ВХОДНЫЕ ДАННЫЕ	Скопировать
<pre>5 2 )) 12 g1))hf)))) 9 gege)))) 14 )aa))b)))))) 1 )</pre>	
ВЫХОДНЫЕ ДАННЫЕ	Скопировать
<pre>Yes No Yes Yes Yes</pre>	

## Разбор задачи А

### 1411A - Игровой чат

Вы должны посчитать количество круглых скобок в конце строки, пусть их  $x$ . Тогда сообщение плохое, если  $x > \frac{n}{2}$ . Обратите внимание, что вы должны делить  $n$  на 2 без округления. Или вместо этого вы можете сравнить  $2 \cdot x$  и  $n$ . Сообщение плохое, если  $2 \cdot x > n$ .

<https://codeforces.com/contest/1411/problem/B>

## В. Справедливые числа

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Назовём натуральное число *справедливым*, если оно делится на каждую из своих ненулевых цифр. Например, число 102 справедливо (так как оно делится и на 1, и на 2), а число 282 — нет, потому что не делится на 8. По данному  $n$  найдите минимальное  $x$ , такое что  $n \leq x$  и  $x$  — справедливо.

### Входные данные

В первой строке содержится  $t$  — количество тестовых случаев ( $1 \leq t \leq 10^3$ ). В каждой из следующих  $t$  строк по одному целому числу  $n$  ( $1 \leq n \leq 10^{18}$ ).

### Выходные данные

Для каждого из  $t$  тестовых случаев в новой строке выведите наименьшее справедливое число, не меньшее  $n$ .

### Пример

входные данные	Скопировать
4 1 282 1234567890 100000000000000000	
выходные данные	Скопировать
1 288 1234568040 100000000000000000	

### Примечание

Пояснения к некоторым тестовым случаям:

- В первом тестовом случае число 1 само по себе является справедливым.
- Во втором тестовом случае число 288 — справедливо (делится и на 2, и на 8). Ни одно число из отрезка [282, 287] не является справедливым, потому что, например, не делится на 8.

## Разбор задачи В

### 1411В - Справедливые числа

Давайте назовём число супер-справедливым, если оно делится на каждую из цифр 1..9. Справедливо заметить, что супер-справедливые числа делятся и на НОК(1..9), равный 2520. Справедливые числа встречаются не реже супер-справедливых, значит можно увеличивать исходное  $n$  на единичку, пока оно не станет справедливым. Перебирать, как мы выяснили, придется немного, а проверять число на справедливость будем, отсоединяя по одной цифре.

<https://codeforces.com/contest/1411/problem/C>

## С. Ладейное поле

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Дана шахматная доска размера  $n \times n$ . Строки и столбцы доски пронумерованы от 1 до  $n$ . Клетка  $(x, y)$  лежит на пересечении столбца номер  $x$  и строки номер  $y$ .

Ладья это шахматная фигура, которая за один ход может переместиться на любое количество клеток по вертикали, либо по горизонтали. На доске размещены  $m$  ладей ( $m < n$ ), которые не бьют друг друга. То есть, нет пары ладей, стоящих на одной вертикали или горизонтали.

За один ход можно сделать ход одной из ладей. То есть, переместить ее на любое количество клеток по вертикали или горизонтали. При этом, после хода ладья снова не должна оказаться под боем других ладей. Какое минимальное количество ходов нужно сделать, чтобы разместить все ладьи на главной диагонали?

Главной диагональю называются клетки  $(i, i)$ , где  $1 \leq i \leq n$ .

### Входные данные

В первой строке дано целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 10^3$ ). Далее дано описание  $t$  тестовых случаев.

В первой строке даны два целых числа  $n$  и  $m$  — размер поля и количество ладей ( $2 \leq n \leq 10^5$ ,  $1 \leq m < n$ ). Следующие  $m$  строк содержат пары целых чисел  $x_i$  и  $y_i$  — позиции ладей,  $i$ -я ладья исходно стоит в клетке  $(x_i, y_i)$  ( $1 \leq x_i, y_i \leq n$ ). Гарантируется, что в исходной расстановке ладьи не бьют друг друга.

Сумма  $n$  по всем тестовым случаям не превышает  $10^5$ .

### Выходные данные

Для каждого из  $t$  тестовых случаев в новой строке выведите наименьшее количество ходов, которые требуются, чтобы поставить все ладьи на главную диагональ.

Можно доказать, что это всегда возможно.

### Пример

входные данные	Скопировать
4 3 1 2 3 3 2 2 1 1 2 5 3 2 3 3 1 1 2 5 4 4 5 5 1 2 2 3 3	
выходные данные	Скопировать
1 3 4 2	

### Примечание

Возможные ходы для первых трех тестовых случаев:

- $(2, 3) \rightarrow (2, 2)$
- $(2, 1) \rightarrow (2, 3)$ ,  $(1, 2) \rightarrow (1, 1)$ ,  $(2, 3) \rightarrow (2, 2)$
- $(2, 3) \rightarrow (2, 4)$ ,  $(2, 4) \rightarrow (4, 4)$ ,  $(3, 1) \rightarrow (3, 3)$ ,  $(1, 2) \rightarrow (1, 1)$

## Разбор задачи С

## 1411C - Ладейное поле

Рассмотрим ладьи как ребра в графе. Позиция  $(x, y)$  будет соответствовать ребру  $(x \rightarrow y)$ . Из условия следует, что из каждой вершины выходит не более одного ребра и входит не более одного ребра. Граф распадается на циклы, пути и петли (ребра вида  $v \rightarrow v$ ).

Что происходит с графом, когда мы передвигаем ладью? Ребро меняет ровно один из концов. Такими операциями мы должны превратить все ребра в петли, при этом должно выполняться ограничение на кол-во ребер входящих и выходящих из вершины. Путь довольно легко превратить в петли, просто начнем с какого-то конца. Цикл же надо сначала превратить в путь, это всегда возможно. Мы потратили только один лишний ход, не сложно убедиться, что это оптимально.

Ответ равен кол-во ладей минус кол-во петель плюс кол-во циклов.

<https://codeforces.com/contest/1411/problem/D>

### D. Грайм Зоопарк

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Сейчас рэп ХХОСа представляет из себя строку из нулей, единиц и знаков вопроса. К сожалению, хейтеры не дремлют. За каждое вхождение подпоследовательности 01 в рэп ХХОСа хейтеры напишут  $x$  гневных комментариев, а за каждое вхождение подпоследовательности 10 будет написано  $y$  гневных комментариев. Вы должны заменить каждый знак вопроса на 0 либо 1, чтобы минимизировать число гневных комментариев, которые получит ХХОС.

Подпоследовательностью строки  $a$  называется строка  $b$ , которая может получиться в результате удаления нескольких символов из строки  $a$ . Два вхождения подпоследовательности считаются разными, если различаются множества позиций оставленных символов.

#### Входные данные

В первой строке записан рэп ХХОСа — строка  $s$  ( $1 \leq |s| \leq 10^5$ ). Во второй строке даны два целых числа  $x$  и  $y$  — количество гневных комментариев, которые ХХОС получит за каждую подпоследовательность 01 и 10, соответственно ( $0 \leq x, y \leq 10^6$ ).

#### Выходные данные

В единственной строке выведите минимальное число гневных комментариев, которые может получить ХХОС.

#### Примеры

входные данные	Скопировать
0?1 2 3	
выходные данные	Скопировать
4	
входные данные	Скопировать
????? 13 37	
выходные данные	Скопировать
0	
входные данные	Скопировать
?10? 239 7	
выходные данные	Скопировать
28	
входные данные	Скопировать
01101001 5 7	
выходные данные	Скопировать
96	

### Примечание

В первом примере одним из оптимальных вариантов замены является 001. Тогда в строке будет 2 подпоследовательности 01 и 0 подпоследовательностей 10. Суммарное количество гневных комментариев равно  $2 \cdot 2 + 0 \cdot 3 = 4$ .

Во втором примере одним из оптимальных вариантов замены является 11111. Тогда в строке будет 0 подпоследовательностей 01 и 0 подпоследовательностей 10. Суммарное количество гневных комментариев равно  $0 \cdot 13 + 0 \cdot 37 = 0$ .

В третьем примере одним из оптимальных вариантов замены является 1100. Тогда в строке будет 0 подпоследовательностей 01 и 4 подпоследовательности 10. Суммарное количество гневных комментариев равно  $0 \cdot 239 + 4 \cdot 7 = 28$ .

В четвертом примере одним из оптимальных вариантов замены является 01101001. Тогда в строке будет 8 подпоследовательностей 01 и 8 подпоследовательностей 10. Суммарное количество гневных комментариев равно  $8 \cdot 5 + 8 \cdot 7 = 96$ .

## Разбор задачи D

### 1411D - Грайм Зоопарк

Рассмотрим два соседних знака вопроса на позициях  $l$  и  $r$  соответственно ( $l < r$ ). Пусть на интервале  $(l, r)$  находится  $c_0$  нулей и  $c_1$  единиц. В случае  $s_l = 0, s_r = 1$  будет написано  $(c_1 + 1) \cdot x + c_0 \cdot x + out = (c_0 + c_1 + 1) \cdot x + out = (r - l) \cdot x + out$  комментариев, где  $out$  - число комментариев за подпоследовательности, хотя бы один элемент которых находится вне  $[l, r]$ . В случае  $s_l = 1, s_r = 0$  получим  $(c_0 + 1) \cdot y + c_1 \cdot y + out = (c_0 + c_1 + 1) \cdot y + out = (r - l) \cdot y + out$  комментариев. Вычтем из первого второе, получим  $(r - l) \cdot (x - y)$ . Это значит следующее: если  $x \geq y$ , всегда выгодно менять 01 на 10, то есть существует такое оптимальное замещение ? на 0 и 1, что первые несколько ? заменяются на 1, а оставшиеся на 0. В случае  $x < y$  аналогично, будет какой-то префикс 0, потом 1. За  $\mathcal{O}(n)$  написать это можно, насчитав, сколько на каждом префиксе ноликов и единичек, и перебирая границу разделения.

<https://codeforces.com/contest/1411/problem/E>

### Е. Ромские цифры

ограничение по времени на тест: 1 секунда  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Ваш друг прислал вам строку  $S$ , состоящую из  $n$  строчных английских букв. Оказывается, это число, записанное в ромской системе счисления. Знания о ромских числах давно утрачены, но сохранился алгоритм перевода ромских чисел в привычные нам. Пронумеруем символы  $S$  от 1 до  $n$  слева направо. Значение числа  $S$  обозначим как  $f(S)$ , оно определяется следующим образом:

- Если  $|S| > 1$ , то выбирается произвольное значение  $m$  ( $1 \leq m < |S|$ ) и считается, что  $f(S) = -f(S[1, m]) + f(S[m + 1, |S|])$ , где  $S[l, r]$  — это подстрока  $S$  с  $l$ -й позиции до  $r$ -й включительно.
- Иначе  $S = c$ , где  $c$  — какая-то буква. В таком случае  $f(S) = 2^{pos(c)}$ , где  $pos(c)$  — это позиция буквы  $c$  в английском алфавите ( $pos(a) = 0, pos(z) = 25$ ).

Обратите внимание, что  $m$  выбирается независимо на каждом шаге.

Друг уверен, что правильно выбирая  $m$  на каждом шаге, можно получить  $f(S) = T$ . Прав ли он?

#### Входные данные

В первой строке даны числа  $n$  и  $T$  ( $2 \leq n \leq 10^5, -10^{15} \leq T \leq 10^{15}$ ).

Во второй строке дана строка  $S$  длины  $n$ , состоящая из строчных английских букв.

#### Выходные данные

Выведите «Yes», если можно получить требуемое значение, иначе выведите «No».

Вы можете выводить каждую букву в любом регистре (строчную или заглавную).

## Примеры

ВХОДНЫЕ ДАННЫЕ	Скопировать
2 -1 ba	
ВЫХОДНЫЕ ДАННЫЕ	Скопировать
Yes	
ВХОДНЫЕ ДАННЫЕ	Скопировать
3 -7 abc	
ВЫХОДНЫЕ ДАННЫЕ	Скопировать
No	
ВХОДНЫЕ ДАННЫЕ	Скопировать
7 -475391 qohshra	
ВЫХОДНЫЕ ДАННЫЕ	Скопировать
Yes	

## Примечание

Во втором примере нельзя получить  $-7$ . Но можно получить, например,  $1$  следующим образом:

1. Сперва выбираем  $m = 1$ , тогда  $f(abc) = -f(a) + f(bc)$
2.  $f(a) = 2^0 = 1$
3.  $f(bc) = -f(b) + f(c) = -2^1 + 2^2 = 2$
4. Итого  $f(abc) = -1 + 2 = 1$

## Разбор задачи E

### 1411E - Ромские цифры

Сперва заметим, что последняя цифра обязательно должна быть взята со знаком плюс, а предпоследняя — со знаком минус. Оказывается, все остальные цифры могут быть взяты с любым знаком. Докажем это.

Пусть мы хотим получить маску  $---++--++---$ . Все минусы слева можно получить, просто расцепляя по одному символу. Осталась маска  $++--++--$ , разобьем ее следующим образом:  $(++--++-)(--)$ . Т.е. оставляем только один минус в последнем отрезке из подряд идущих минусов. Поменяем знаки:  $(--++--)(--)$ . Свели к задаче меньше. Действуя подобным образом, мы в конце придем к строкам вида  $+$ .

Теперь задача свелась к тому, можно ли набрать число  $X$  используя первые  $n - 2$  букв. Так как веса предметов - степени двойки, их можно набирать жадно.

Бонус. Сконструируйте бинарное дерево, соответствующее маске, за линию.

<https://codeforces.com/contest/1411/problem/E>

## Ф. Тернистый путь

ограничение по времени на тест: 2 секунды  
ограничение по памяти на тест: 256 мегабайт  
ввод: стандартный ввод  
вывод: стандартный вывод

Легенда гласит, что в Ханойском храме хранится перестановка чисел от 1 до  $n$ . Перед храмом в линию лежат  $n$  разноцветных камней. Жрецы могут проводить следующую операцию над камнями: выбрать позицию  $i$  ( $1 \leq i \leq n$ ) и циклически сдвинуть камни на позициях  $i, p[i], p[p[i]], \dots$ . То есть, камень с позиции  $i$  перейдет на позицию  $p[i]$ , камень с позиции  $p[i]$  перейдет на позицию  $p[p[i]]$ , и т.д., на позицию  $i$  перейдет камень с позиции  $j$ , такой что  $p[j] = i$ .

Каждый день жрецы обязаны получать новую расстановку камней, используя произвольное количество этих операций. Когда все возможные расстановки будут получены, наступит конец света. Вам стало интересно, что если бы перед самым началом можно было бы поменять местами некоторые элементы перестановки? Сколько дней бы просуществовал мир?

Вы хотите за минимальное количество обменов двух элементов, получить перестановку, которая позволит миру просуществовать как можно дольше.

Две расстановки камней, считаются различными, если существует позиция  $i$  такая, что цвета камней на этой позиции в расстановках различаются.

### Входные данные

В первой строке дано целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 10^3$ ). Далее дано описание тестовых случаев.

В первой строке дано целое число  $n$  — длина перестановки ( $3 \leq n \leq 10^6$ ). Во второй строке находится  $n$  целых чисел  $p_1, \dots, p_n$  — перестановка ( $1 \leq p_i \leq n$ ). Гарантируется, что  $p$  является перестановкой.

Сумма  $n$  по всем тестовым случаям не превышает  $10^6$ .

### Выходные данные

Для каждого из  $t$  тестовых случаев в новой строке выведите два целых числа: наибольшее возможное количество дней, которые может просуществовать мир, по модулю  $10^9 + 7$  и минимальное количество необходимых для этого обменов.

### Примеры

<b>входные данные</b>	<a href="#">Скопировать</a>
<pre>3 3 2 3 1 3 2 1 3 3 1 2 3</pre>	
<b>выходные данные</b>	<a href="#">Скопировать</a>
<pre>3 0 3 1 3 2</pre>	
<b>входные данные</b>	<a href="#">Скопировать</a>
<pre>5 4 2 3 4 1 4 2 3 1 4 4 2 1 4 3 4 2 1 3 4 4 1 2 3 4</pre>	
<b>выходные данные</b>	<a href="#">Скопировать</a>
<pre>4 0 4 1 4 0 4 1 4 2</pre>	

### Примечание

Обозначим цвета камней буквами. Пояснения для первых двух тестовых случаев первого примера:

1. Используя перестановку [2, 3, 1], из ABC можно дополнительно получить расстановки CAB и BCA. Что уже является максимально возможным результатом.
2. Используя перестановку [2, 1, 3], из ABC можно получить только BAC. Как мы видели в предыдущем примере, две расстановки не являются максимально возможным количеством для  $n = 3$ . Для получения оптимальной перестановки, например, можно поменять местами 1 и 3, чтобы получить перестановку [2, 3, 1].

## Разбор задачи F

### 1411F - Тернистый путь

Задача сводится к тому, что надо из массива целых чисел, используя операции объединения и разъединения, получить массив из троек и остатка (2 или 2+2 или 4). Помогает считать, что все операции объединения делаются до операций разъединения.

Чтобы решить задачу, можно перебрать из каких элементов массива вычитается остаток, тогда остальные операции выполняются жадно.

Бонус. Дано  $k$ . Надо получить массив из  $k$ , используя эти операции. Считаем, что сумма элементов массива делится на  $k$  без остатка. Это можно представить как минимальное покрытие гиперграфа ребрами с весами = (кол-во вершин - 1) + (сумма элементов /  $k - 1$ ). Существует ли полиномиальное решение ( $k$  является параметром)?

<https://codeforces.com/contest/1411/problem/G>

### G. Нет игры - нет жизни

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Рассмотрим следующую игру Алисы и Боба на ориентированном ациклическом графе. Каждая вершина может содержать произвольное количество фишек. Алиса и Боб совершают ходы по очереди. Первой ходит Алиса. Ход состоит в том, чтобы передвинуть ровно одну фишку по какому-то ребру, исходящему из вершины, в которой сейчас лежит фишка, в конец этого ребра. Проигрывает тот, кто не может сделать ход. Оба играют оптимально.

Рассмотрим следующий процесс, происходящий каждую секунду на данном графе с  $n$  вершинами:

1. Целое число  $v$  выбирается случайно и равновероятно из  $[1, n + 1]$ .
2. Если  $v \leq n$ , в  $v$ -ю вершину графа добавляется фишка, а процесс переходит к шагу 1.
3. Если  $v = n + 1$ , Алиса и Боб играют в описанную выше игру на данном графе с текущим расположением фишек, определяется победитель этой игры. После чего, процесс завершается.

Найдите вероятность победы Алисы. Можно показать, что ответ можно представить в виде  $\frac{P}{Q}$ , где  $P$  и  $Q$  — взаимно простые целые числа,  $Q \not\equiv 0 \pmod{998\,244\,353}$ . Выведите значение  $P \cdot Q^{-1} \pmod{998\,244\,353}$ .

#### Входные данные

В первой строке даны два целых числа  $n$  и  $m$  — количество вершин и ребер в графе ( $1 \leq n \leq 10^5$ ,  $0 \leq m \leq 10^5$ ).

Затем следует  $m$  строк, в  $i$ -й из которых содержатся два целых числа  $u_i$  и  $v_i$  — начало и конец  $i$ -го ребра ( $1 \leq u_i, v_i \leq n$ ). Гарантируется, что граф является ациклическим.

#### Выходные данные

В единственной строке выведите вероятность победы Алисы по модулю 998 244 353.



## Примеры

входные данные	Скопировать
1 0	
выходные данные	Скопировать
0	
входные данные	Скопировать
2 1 1 2	
выходные данные	Скопировать
332748118	
входные данные	Скопировать
5 5 1 4 5 2 4 3 1 5 5 4	
выходные данные	Скопировать
931694730	

## Разбор задачи G

### 1411G - Нет игры - нет жизни

Кто выиграл при данной конфигурации фишек зависит только от ксора значений Гранди вершин по всем находящимся на доске фишкам. Заметим, что каждое значение Гранди  $\leq \sqrt{m}$ , а значит их ксор не превосходит 512. Пусть  $P_v$  это вероятность победы Алисы, если текущее состояние имеет ксор  $v$ .

$$P_v = \sum P_{to} \cdot \text{prob}(v \rightarrow to) + [v \neq 0] \cdot \frac{1}{n+1}$$

Второе слагаемое - мы получили  $n + 1$  и процесс завершился. Понятно, что  $\text{prob}(v \rightarrow to) = \frac{\text{cnt}[v \oplus to]}{n+1}$ , где  $\text{cnt}[x]$  это количество вершин со значением Гранди равным  $x$ . Теперь у нас есть система из 512 уравнений с неизвестными  $P_v$ , можем решить её методом Гаусса. Ответ в  $P_0$ . Доказательство того, что Гаусс не сломается по пути, оставим читателю в качестве упражнения.

Также существует решение, использующее преобразование Адамара.