

# Любитель мозаик

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

У вас есть  $n$  белых и  $m$  чёрных камешков. Используя **все** камешки, необходимо собрать **красивую мозаику**.

Мозаика считается **красивой**, если выполняются следующие условия:

- она имеет квадратную форму (то есть состоит из  $k \times k$  камешков для некоторого целого  $k > 0$ );
- каждая строка в ней является **палиндромом**, то есть читается одинаково слева направо и справа налево.

Определите, можно ли составить **красивую мозаику**, используя **все** камешки.

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 100$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Далее следуют  $t$  строк, каждая из которых содержит два целых числа  $n$  и  $m$  ( $0 \leq n, m \leq 2500$  и  $n + m \leq 2500$ ) — количество белых и чёрных камешков соответственно.

## Формат выходных данных

Для каждого набора входных данных выведите «YES», если можно составить **красивую мозаику**, и «NO» в противном случае. Выводите ответы в том же порядке, в котором заданы входные наборы данных.

## Пример

стандартный ввод	стандартный вывод
3	YES
4 5	NO
2 3	YES
8 8	

## Замечание

В первом примере можно составить мозаику  $3 \times 3$ . Например, одна из корректных раскладок:

W	B	W
B	W	B
W	B	W

Во втором примере нельзя составить квадратную мозаику, используя все камешки. В третьем примере из 8 белых и 8 чёрных можно собрать мозаику  $4 \times 4$ .

# Подмозаики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Начинающий мозаичист Марат на днях составил две черно-белые мозаики  $a$  и  $b$  на прямоугольных холстах одинакового размера.

Каждая мозаика состоит из  $n$  строк и  $m$  столбцов. На пересечении  $i$ -й строки и  $j$ -го столбца находится квадратный элемент мозаики, покрашенный в цвет  $a_{i,j}$  у первой мозаики и в цвет  $b_{i,j}$  у второй.

Марат хочет рассмотреть все такие пары подмозаик, что:

1. Первая подмозаика — это непрерывный подпрямоугольник в первой мозаике;
2. Вторая подмозаика — это непрерывный подпрямоугольник во второй мозаике;
3. Индексы углов подмозаик совпадают;
4. Значения цветов в обоих подмозаиках во всех позициях совпадают.

Обратите внимание, что в каждой паре подмозаики имеют одинаковую площадь. Марат интересуется, насколько большая площадь может быть.

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 10^3$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 1000$ ) — размеры мозаик.

Каждая из следующих  $n$  строк содержит  $m$  символов — описание мозаики  $a$ . На  $i$ -й строке в  $j$ -м столбце находится число  $a_{i,j}$  ( $a_{i,j} \in \{0, 1\}$ ) — описание цвета очередного элемента первой мозаики.

Каждая из следующих  $n$  строк содержит  $m$  символов — описание мозаики  $b$ . На  $i$ -й строке в  $j$ -м столбце находится число  $b_{i,j}$  ( $b_{i,j} \in \{0, 1\}$ ) — описание цвета очередного элемента первой мозаики.

Гарантируется, что сумма по всем  $n \times m$  не превосходит  $10^6$ .

## Формат выходных данных

Для каждого набора входных данных выведите одно целое число — ответ на задачу.

## Пример

стандартный ввод	стандартный вывод
5	20
4 5	0
00110	0
10101	1
01010	4
01010	
00110	
10101	
01010	
01010	
4 5	
00000	
00000	
00000	
00000	
11111	
11111	
11111	
11111	
1 1	
0	
1	
1 1	
1	
1	
5 3	
110	
110	
010	
010	
010	
111	
111	
101	
101	
101	

# Коридор из мозаики

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

В Турции очень много древних достопримечательностей. Приехав в эту страну, Вы первым делом решили записаться на экскурсию в какой-нибудь храм. На месте вы увидели длинный коридор, пол которого состоял из больших фрагментов мозаики, выстроенных в линию.

Длина шага позволяла Вам двигаться на соседний фрагмент или через один. Достигнув конца этого коридора, Вы посчитали число фрагментов — их было  $n$ .

Под впечатлением экскурсии Вы много думали и задались разными вопросами. Один из них: а как происходит реставрация? Логичное предположение, что иногда некоторые фрагменты достают и заменяют отреставрированными. А их на месте на время реставрации ставят знак, запрещающий туда наступать. Получается настоящая полоса препятствий!

И дальше Вы начали думать: а существует ли такая конфигурация из реставрируемых фрагментов, что количество различных способов пройти по коридору, двигаясь только вперёд на соседний фрагмент или через один, равняется  $k$ , при этом отправлять на реставрацию фрагменты с номерами 1 и  $n$  нельзя. В уме решить такую задачу не смогли и решили написать программу, которая даст ответ на этот вопрос.

## Формат входных данных

Единственная строка входных данных содержит два целых числа,  $2 \leq n \leq 10^4$  — количество клеток и  $1 \leq k \leq 10^{18}$  — желаемое количество способов пройти по коридору из мозаики.

## Формат выходных данных

В первой строке выведите одно целое число  $l$  — количество одновременно реставрируемых фрагментов. Если конфигурации не существует, то выведите  $-1$ .

Если искомая конфигурация достижима, то во второй строке выведите  $l$  целых чисел в возрастающем порядке, разделённых пробелом:  $a_1, \dots, a_l$  — номера фрагментов, которые необходимо отправить на реставрацию, чтобы число способов добраться до конца коридора равнялось  $k$ .

## Примеры

стандартный ввод	стандартный вывод
9 10	1 6
4 3	0

# Восстановление мозаики

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

## Это интерактивная задача.

Вы ведёте археологические раскопки и только что обнаружили  $N$  частей древней мозаики. Вы пронумеровали эти части от 1 до  $N$  и хотите выложить их в один ряд на позиции от 1 до  $N$ . Правда, вы пока не уверены, какую из частей нужно разместить в каждой из позиций.

Вы решили проконсультироваться с коллегами из местного исторического общества. К сожалению, средств на вашем спутниковом телефоне хватит только на 10 сообщений.

В одном сообщении вы можете отправить фото одной или нескольких частей мозаики, а в ответ коллеги отправят вам номера позиций, в которых эти части следует разместить. Порядок фото при отправке нарушается, поэтому коллеги всегда будут присылать вам список позиций в порядке возрастания.

Восстановите мозаику.

## Протокол взаимодействия

Первая строка входных данных содержит одно целое число  $N$  ( $1 \leq N \leq 500$ ) — количество частей мозаики.

Далее вы можете сделать не более 10 запросов. Каждый запрос должен быть одного из двух видов:

- Если вы хотите проконсультироваться с коллегами, выведите слово **CHECK**. Далее в этой строке выведите  $K$  ( $1 \leq K \leq N$ ) различных целых чисел  $A_i$  ( $1 \leq A_i \leq N$ ) — номера частей мозаики, фото которых вы отправляете коллегам.

После этого запроса ваша программа должна ввести  $K$  различных целых чисел  $P_i$  ( $1 \leq P_i \leq N$ ) — номера позиций, на которых должны располагаться описанные части, в порядке возрастания.

- Если вы определили правильный порядок частей, выведите слово **RESULT**. Далее в этой строке выведите  $N$  различных целых чисел  $A_j$  ( $1 \leq A_j \leq N$ ) — номера частей мозаики, которые должны быть размещены в позициях 1, 2, ...,  $N$ .

Этот запрос должен быть последним, после него ваша программа должна завершить работу.

В конце каждого запроса не забудьте вывести перевод строки и сбросить буфер вывода:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `stdout.flush()` в Python;
- Для других языков см. документацию.

## Примеры

стандартный ввод	стандартный вывод
3  1 2  2 3	CHECK 1 2  CHECK 2 3  RESULT 1 2 3
4  2 3 4  1 2  1 4	CHECK 1 2 3  CHECK 2 4  CHECK 4 3  RESULT 4 2 1 3

# Закупаем Лукум

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

Недавно в Яндекс Лавке появилась возможность заказать Рахат Лукум — одну из самых популярных сладостей, придуманных в Османской империи в конце XVIII века в Стамбуле. Всего на продажу выставлены  $n$  различных позиций лукума, пронумерованных от 1 до  $n$ , и, к счастью, некоторые было решено раздать любимым участникам Кубка Яндекса.

Известно, что будет  $k$  различных независимых друг от друга акций, и  $i$ -я акция будет покрывать все позиции с номерами в диапазоне от  $l_i$  до  $r_i$  и при этом существовать с вероятностью  $p_i/q_i$ . Иными словами, вероятность существования одной акции не зависит от существования другой, и если акция существует, то она применяется ко всем товарам в указанном диапазоне номеров.

Теперь же Вас попросили для каждой позиции определить вероятность того, что к ней будет применена ровно одна акция. Для каждой позиции выведите искомую вероятность по модулю  $10^9 + 7$ .

Вероятность можно выразить в виде несократимой дроби  $\frac{x}{y}$ . Вы должны вывести значение  $x \cdot y^{-1} \bmod (10^9 + 7)$ , где  $y^{-1}$  — целое число, такое, что  $y \cdot y^{-1} \bmod (10^9 + 7) = 1$ .

## Формат входных данных

Первая строка ввода содержит два целых числа  $n$  и  $k$  ( $1 \leq n, k \leq 2 \cdot 10^5$ ) — количество предложений и количество акций.

$i$ -я из следующих  $k$  строк ввода содержит по 4 целых числа, описывающих  $i$ -ю акцию  $l_i, r_i, p_i, q_i$  ( $1 \leq l_i \leq r_i \leq n, 1 \leq p_i < q_i < 10^9 + 7$ )

## Формат выходных данных

Выведите единственную строку, содержащую  $n$  целых чисел, где  $i$ -е число представляет собой вероятность по модулю  $10^9 + 7$  того, что  $i$ -е предложение будет покрыто ровно одной акцией.

## Пример

стандартный ввод	стандартный вывод
5 3 1 3 1 2 2 4 1 3 3 4 5 6	5000000004 5000000004 3611111114 6111111116 0

# Композиция мозаик

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

Гуляя по туристическим местам, Вы увидели в сувенирном магазине интересную игру, где надо из небольших фрагментов собирать мозаику. Каждый фрагмент промаркирован цифрой от 0 до 9.

Вы закупили их огромное количество. Таким образом, можно считать, что экземпляров каждого фрагмента у вас бесконечное множество.

Добравшись до отеля, Вы решили собрать  $k$  **красивых** мозаик, каждая из которых представляет собой некоторое количество фрагментов, выложенных в ряд. Мозаика считается **красивой**, если число, образованное цифрами с фрагментов мозаики, не содержит лидирующих нулей. **Красотой мозаики** является число, полученное путём последовательного выписывания цифр соответствующих фрагментов в десятичное число.

Собрав множество красивых мозаик, Вы решили объединить их в композицию и выложили их в один ряд. **Красотой композиции** решили считать число, полученное выписыванием цифр всех фрагментов мозаик.

В этот момент Вы задались вопросом, какую максимальную **красоту композиции** вы можете получить из мозаик, сумма красоты которых равна  $n$ .

## Формат входных данных

Единственная строка входных данных содержит два целых числа  $1 \leq n \leq 10^{3000}$ ,  $1 \leq k \leq 2000$  — сумма **красот мозаик** и число мозаик в композиции, соответственно.

## Формат выходных данных

В единственной строке выведите  $k$  целых чисел, разделённых пробелом — значения **красот мозаик**, входящих в композицию, расположенных слева направо в композиции с максимальной красотой.

## Примеры

стандартный ввод	стандартный вывод
13 2	3 10
4 3	4 0 0



# Зейтин и красивая мозаика

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

Стамбульский горилленок Зейтин нашёл мозаику из  $n$  фрагментов. Фрагменты расположены в ряд и имеют целочисленные цвета  $a_1, a_2, \dots, a_n$ .

Зейтин не любит одинокие простые делители. Он считает фрагмент мозаики цвета  $x$  *красивым* только если для каждого простого  $p$  верно, что если  $x$  делится на  $p$  без остатка, то  $x$  делится на  $p^2$  без остатка. Например, фрагменты мозаики с цветами 4, 8, 200 — *красивые*, а фрагменты с цветами 2, 13, 75 — нет.

Подотрезок мозаики  $a_l, a_{l+1}, \dots, a_r$  считается *красивым* только если **каждый** его фрагмент *красивый*.

НОД подотрезка мозаики  $a_l, a_{l+1}, \dots, a_r$  — это такое **наибольшее** целое число, которое является делителем цвета каждого фрагмента мозаики  $a_l \dots a_r$ .

Помогите Зейтину посчитать сумму НОД всех *красивых* подотрезков мозаики. Если *красивых* подотрезков мозаики нет, сумма считается равной нулю.

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество элементов в мозаике.

Вторая строка каждого набора входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^7$ ) — цвета фрагментов мозаике.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ .

## Формат выходных данных

Для каждого набора входных данных выведите единственное целое число — ответ на задачу.

## Пример

стандартный ввод	стандартный вывод
4	0
1	10000000
2	866
5	30125379
40 10000000 2 13 228	
6	
25 8 2 200 200 216	
7	
8869743 26569 120125 4959529 9150625 1865956 5132808	

# Мозаичный шифр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

## Это задача с двойным запуском.

Скоро вам предстоит поездка в Турцию. Вы очень хотите проникнуться атмосферой, и поэтому всю ночь смотрели познавательные телепередачи про историю Турции. Вы узнали, что раньше там жили лучшие учёные своего времени.

В одном из сюжетов рассказывалось, что учёные умели кодировать бинарные числа в виде мозаик! А так как мозаики — это реальные объекты, то они изобрели способ, который позволял сделать это с ограничением на число фрагментов в мозаике.

Вы знаете, что древние учёные брали бинарную строку длины  $n$  и кодировали её с помощью правильной скобочной последовательности длины не более  $n + 70$ . Выкладывали мозаику из фрагментов двух цветов, где открывающей скобке был сопоставлен чёрный фрагмент, а закрывающей — белый. Эту мозаику можно было передать на другой конец империи, и местный учёный, знающий шифр, мог декодировать в изначальную строку.

Вам стало интересно, а как можно реализовать такой шифр? Вы уже написали программу для проверки корректности и тесты. Осталось придумать сам алгоритм шифрования!

## Протокол взаимодействия

Первая строка входных данных содержит одно целое число  $R$  ( $R \in \{1, 2\}$ ) — номер текущего запуска.

### 1 Первый запуск

Если  $R = 1$ , то вторая и последующие строки входных данных описываются следующим образом.

Вторая строка входных данных содержит одно целое число  $t$  ( $1 \leq t \leq 10^5$ ) — количество независимых тестовых случаев.

Каждая из следующих  $t$  строк содержит бинарную строку  $s_i$ . Гарантируется, что  $\sum_{i=1}^t |s_i| \leq 10^6$ , где  $|s_i|$  обозначает длину  $i$ -й строки.

Для каждого тестового случая выведите в отдельной строке корректную ПСП. Длина ответа для  $i$ -го теста не должна превышать  $|s_i| + 70$ . Ответы необходимо выводить в том же порядке, в котором поступают входные строки.

### 2 Второй запуск

Если  $R = 2$ , то вторая и последующие строки входных данных описываются следующим образом.

Вторая строка входных данных содержит одно целое число  $t$  ( $1 \leq t \leq 10^5$ ) — количество независимых тестовых случаев.

В каждой из следующих  $t$  строк подается ПСП — одна из тех последовательностей, которые ваше решение выдало на первом запуске. Обратите внимание, что порядок ПСП может отличаться от порядка их вывода на первом запуске.

Для каждой ПСП выведите исходную бинарную строку, которая была закодирована на первом запуске. Строки следует выводить в порядке появления соответствующих ПСП во входных данных.

## Примеры

стандартный ввод	стандартный вывод
1 2 101010 000111	((()()())) ((((())))())
2 2 ((((())))()) (()()())	000111 101010

## Замечание

Правильная скобочная последовательность (ПСП) — это последовательность из открывающих и закрывающих круглых скобок, в которой между скобками можно расставить символы «+» и 1 так, чтобы получившееся выражение стало корректным арифметическим.

# Из Стамбула на Манхэттен

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

В Стамбуле растет множество разных деревьев: кипарисы, платаны, магнолии, сосны и другие. Бывают так же и довольно необычные: например, встречается тот вид деревьев, что называют ненаправленным связным ациклическим графом.

Арслан смог договориться о поставке дерева последнего вида на Манхэттен. Разумеется, на Манхэттене есть свои правила того, как нужно размещать дерево. Они описаны ниже.

- Размещение дерева на Манхэттене происходит в  $k$ -мерном пространстве с манхэттенской метрикой.
- Каждой вершине исходного дерева однозначно сопоставляется некоторая точка в этом пространстве. Для вершины  $v$  будем обозначать эту точку как  $p(v)$ .
- Между любыми двумя вершинами  $u$  и  $v$  расстояние в исходном дереве **совпадает** с расстоянием между точками  $p(v)$  и  $p(u)$  в заданном пространстве.
- Число  $k$  должно быть **минимально** возможным (чтобы дерево не занимало слишком много места!).

Несмотря на договоренность о поставке, Арслан немного не подумал о том, как он будет выполнять эти странные правила размещения дерева, поэтому просит Вас о помощи!

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $1 \leq t \leq 200$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 100$ ) — количество вершин в дереве.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ ) — описание очередного ребра. Гарантируется, что множество ребер образует дерево.

## Формат выходных данных

Для каждого набора входных данных выведите следующее.

В первой строке выведите одно целое число  $k$  ( $1 \leq k \leq 10^3$ ) — минимальная размерность манхэттенского пространства.

В каждой из следующей  $n$  строк выведите  $k$  целых чисел, где на  $v$ -й строке должны быть выведены  $p(v)_1, p(v)_2, \dots, p(v)_k$  — координаты точки  $p(v)$  ( $-10^6 \leq p(v)_i \leq 10^6$ )

Если существует несколько решений, выведите любое.

## Пример

стандартный ввод	стандартный вывод
5	1
2	0
1 2	1
5	2
1 2	1 0
1 3	2 0
1 4	1 -1
1 5	1 1
10	0 0
1 2	3
2 3	4 0 0
2 4	3 0 0
4 5	3 0 1
4 6	2 0 0
6 7	2 1 0
6 10	1 0 0
7 8	1 -1 0
7 9	1 -2 0
9	1 -1 -1
1 2	0 0 0
1 3	3
1 4	3 0 0
4 5	4 0 0
4 6	3 0 1
6 7	2 0 0
6 8	2 0 -1
6 9	1 0 0
1	1 -1 0
	1 1 0
	0 0 0
	1
	0

# За нами следят

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Как всем известно, за нами следят. Инопланетяне. Прямо из космоса. И чтобы не остаться в стороне, хорошо было бы внести вклад в историю, которую пишут внеземные разумы о нашей планете, прилагая к ней различные снимки. Ну или хотя бы засветиться в сюжете на Humanoid-TV. Поэтому ArgovDI, как вождь племени, решил украсить небольшой кусочек своей земли, представляющий поле размерами  $n$  на  $m$  клеток, мозаикой, выложенной из красивых камней. Каждый из них занимает ровно одну клетку поля. На поле, как можно догадаться, сейчас нет ни одного камня.

В его племени есть  $k$  семей, каждая из которых готова предоставить по такому случаю либо один свой камушек (их самую лучшую работу), который необходимо разместить в точности на  $x_i$ -й клетке сверху и  $y_i$ -й клетке слева, либо остальные (пусть они и не такие красивые)  $l_i$  камней, но свою лучшую работу оставить себе. Стоит уточнить, что каждая семья очень хочет поучаствовать, поэтому не может не выставлять ничего. В случае выбора не в сторону самого красивого камня все остальные  $l_i$  камней должны быть расставлены.

Так как ArgovDI в душе еще и дизайнер, он не хочет, чтобы мозаика получилась слишком скучной. Под скучностью он понимает максимум среди следующих величин по всем строкам: количество клеток от самого левого камня до самого правого (включая клетки под ними самими) в рамках одной строки.

ArgovDI, к сожалению, сильно занят вопросами воспитания и развития своего племени, поэтому попросил Вас помочь ему найти способ выбрать камни таким образом, чтобы скучность мозаики была минимально возможной.

## Формат входных данных

Первой строка содержит три целых числа  $1 \leq n, m, k \leq 10^5$ ;  $nm \leq 10^5$  — количество строк в поле, количество столбцов, количество племен соответственно.

Следующая строка содержит  $2k$  целых чисел, записанных через пробел:  $x_1, y_1, x_2, y_2, \dots, x_k, y_k$  — список координат, куда необходимо поместить лучшую работу  $i$ -й семьи ( $1 \leq x_i \leq n, 1 \leq y_i \leq m$ ).

Каждая из следующих  $k$  строк содержит одно целое число  $l_i$  ( $0 \leq l_i < 10^5$ ) — количество камней, которое необходимо расставить в качестве альтернативы лучшей работы семьи. Далее на этой строке заданы  $l_i$  координат:  $X_{i,1}, Y_{i,1}, \dots, X_{i,l_i}, Y_{i,l_i}$ . Все числа на строке разделены пробелом.

Гарантируется, что все точки, предоставленные во входных данных, попарно различны.

## Формат выходных данных

В первой строке выведите единственное число — минимальную возможную скучность мозаики.

Во второй строке выведите целое число  $r$ , а затем, через пробел,  $r$  целых чисел  $i_1, \dots, i_r$  — номера семей, которым необходимо оставить своё лучшее творение себе.

## Пример

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

# Симметрия мозаики

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

Вы очень прониклись древним искусством мозаики. Сделали множество своих творений и теперь хотите научиться их сравнивать.

Для этого каждому фрагменту мозаики Вы сопоставили маленькую латинскую букву. Какое же счастье, что различных фрагментов было не более 26. Таким образом, каждую мозаику можно представить в виде строки  $s$ .

Красотой такой строки является количество пар подотрезков таких, что если развернуть сначала первый подотрезок, а потом второй, то строка не изменится. Так как это число может быть большим, выведите его по модулю 998244353.

Приведем более формальное объяснение. Пусть  $s = s_1s_2\dots s_n$  — строка длины  $n$ . Тогда для  $1 \leq l \leq r \leq n$  обозначим за  $F(s, l, r)$  строку, в которой развернули подотрезок  $[l, r]$ :  $F(s, l, r) = s_1s_2\dots s_{l-1}s_rs_{r-1}\dots s_{l+1}s_ls_{r+1}s_{r+2}\dots s_n$ . Вам нужно посчитать количество пар подотрезков  $([l_1, r_1], [l_2, r_2])$ , таких что  $F(F(s, l_1, r_1), l_2, r_2) = s$ .

Обратите внимание, что порядок подотрезков в паре имеет значение.

## Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число  $t$  ( $t \leq 10^5$ ) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая и единственная строка каждого набора входных данных содержит строку  $s$ . Строка состоит из латинских букв в нижнем регистре.

Гарантируется, что сумма длин строк не превосходит  $10^6$ .

## Формат выходных данных

Для каждого набора входных данных выведите одно целое число — красоту мозаики по модулю 998244353.

## Пример

стандартный ввод	стандартный вывод
7	9
aa	36
aaa	100
aaaa	57
aboba	405
abbaccabba	385
abbracaddabra	338
mozaististanbul	