

В. Перестановка

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Перестановкой p называется упорядоченный набор чисел p_1, p_2, \dots, p_n , состоящий из n различных целых положительных чисел, каждое из которых не больше чем n . Число n будем называть длиной перестановки p_1, p_2, \dots, p_n .

У Семена есть целое положительное число n и целое неотрицательное число k такое, что $2k \leq n$. Помогите ему найти перестановку a длины $2n$ такую, что выполнено равенство: $\sum_{i=1}^n |a_{2i-1} - a_{2i}| - \left| \sum_{i=1}^n a_{2i-1} - a_{2i} \right| = 2k$.

Входные данные
В первой строке записаны два целых числа n и k ($1 \leq n \leq 50000, 0 \leq 2k \leq n$).

Выходные данные
Выведите $2n$ целых чисел a_1, a_2, \dots, a_{2n} — искомую перестановку a . Гарантируется, что решение существует. Если существует несколько решений, разрешается вывести любое.

Примеры

входные данные	Скопировать
1 0	
выходные данные	Скопировать
1 2	
входные данные	Скопировать
2 1	
выходные данные	Скопировать
3 2 1 4	
входные данные	Скопировать
4 0	
выходные данные	Скопировать
2 7 4 6 1 3 5 8	

Примечание
Под записью $|x|$ следует понимать абсолютную величину или модуль числа x .

В первом примере $|1 - 2| - |1 - 2| = 0$.
Во втором примере $|3 - 2| + |1 - 4| - |3 - 2 + 1 - 4| = 1 + 3 - 2 = 2$.
В третьем примере $|2 - 7| + |4 - 6| + |1 - 3| + |5 - 8| - |2 - 7 + 4 - 6 + 1 - 3 + 5 - 8| = 12 - 12 = 0$.

При поддержке



В. Майк и весёлая игра

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Майк с некоторыми мишками играют в веселую игру. Майк — судья. Все медведи, кроме Майка, стоят на клетчатом поле размера $n \times m$, в каждой клетке стоит ровно по медведю. Обозначим медведя на пересечении столбца номер j и строки номер i как (i, j) . Майк держит руки на ушах (так как он судья), а каждый медведь стоящий на поле, закрывает лапами рот или глаза.



Медведи разыгрывают q раундов. В каждом раунде Майк выбирает медведя (i, j) и говорит ему поменять своё состояние, то есть, если медведь закрывает лапами рот, то он должен переместить лапы на глаза, а в противном случае — закрыть лапами рот. После этого Майк хочет знать счёт медведей.

Счет медведей — это максимальное по всем строкам количество стоящих подряд медведей с лапами на глазах в этой строке.

Медведи ленивые, поэтому Майк попросил Вас помочь ему. Для каждого раунда, назовите ему счет медведей после того, как поменет своё состояние выбранный в этом раунде медведь.

Входные данные

В первой строке находится три целых числа, n , m и q ($1 \leq n, m \leq 500$ и $1 \leq q \leq 5000$).

В следующих n строках задано описание поля. Каждая строка состоит из m целых чисел, разделенных пробелами. Каждое из этих чисел равно либо 0 (закрывается рот), либо 1 (закрываются глаза).

В следующих q строках записана информация о раундах. В каждой из них записано два целых числа, i и j ($1 \leq i \leq n$ and $1 \leq j \leq m$), номер строки и столбца медведя, меняющего состояние.

Выходные данные

После каждого раунда выведите текущий счет медведей.

Примеры

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

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Наша Таня громко плачет

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Сейчас она на самом деле не плачет. Но заплачет, если вы не решите эту задачу.

Даны числа n, k, A, B . Есть число x , которое изначально равно n . Вы можете выполнять 2 типа операций:

- 1. Уменьшить x на 1, заплатив A монет.
- 2. Поделить x на k , заплатив B монет. Эту операцию можно выполнять, только если x делится на k .

Какое минимальное количество монет необходимо заплатить для того, чтобы получить $x = 1$?

Входные данные

В первой строке содержится целое число n ($1 \leq n \leq 2 \cdot 10^9$).

Во второй строке содержится целое число k ($1 \leq k \leq 2 \cdot 10^9$).

В третьей строке содержится целое число A ($1 \leq A \leq 2 \cdot 10^9$).

В четвёртой строке содержится целое число B ($1 \leq B \leq 2 \cdot 10^9$).

Выходные данные

Выведите единственное число — минимальное количество монет, которое необходимо заплатить для того, чтобы получить $x = 1$.

Примеры

входные данные	Скопировать
9 2 3 1	
выходные данные	Скопировать
6	

входные данные	Скопировать
5 5 2 20	
выходные данные	Скопировать
8	

входные данные	Скопировать
19 3 4 2	
выходные данные	Скопировать
12	

Примечание

В первом тестовом примере оптимальная стратегия такова:

- Уменьшить x на 1 ($9 \rightarrow 8$) за 3 монеты.
- Уменьшить x в 2 раза ($8 \rightarrow 4$) за 1 монету.
- Уменьшить x в 2 раза ($4 \rightarrow 2$) за 1 монету.
- Уменьшить x в 2 раза ($2 \rightarrow 1$) за 1 монету.

Суммарная стоимость — 6 монет.

Во втором тестовом примере оптимальная стратегия — 4 раза уменьшить x на 1, потратив 8 монеты.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Пройденный путь

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Бортовой компьютер автомобиля Поликарпа зафиксировал, что его скорость в начале некоторого участка пути была равна v_1 метров в секунду, а в конце — v_2 метров в секунду. Известно, что этот участок пути занял ровно t секунд движения.

Считая, что в каждую из секунд скорость постоянна, а между секундами скорость может мгновенно изменяться не более чем на d метров в секунду по абсолютной величине (то есть разность скоростей в любые две соседние секунды по модулю не превосходит d), найдите возможную максимальную длину участка пути в метрах.

Входные данные

В первой строке записаны два целых числа v_1 и v_2 ($1 \leq v_1, v_2 \leq 100$) — скорости в метрах в секунду в начале участка и в его конце соответственно.

Во второй строке записаны два целых числа t ($2 \leq t \leq 100$) — время движения по участку в секундах, d ($0 \leq d \leq 10$) — максимальное значение, на которое может измениться скорость между соседними секундами.

Гарантируется, что существует способ проехать участок так, что:

- скорость в первую секунду будет равна v_1 ,
- скорость в последнюю секунду будет равна v_2 ,
- абсолютная величина разности скоростей в любые две соседние секунды не превосходит d .

Выходные данные

Выведите максимально возможную длину участка пути в метрах.

Примеры

входные данные	Скопировать
5 6 4 2	
выходные данные	Скопировать
26	

входные данные	Скопировать
10 10 10 0	
выходные данные	Скопировать
100	

Примечание

В первом примере последовательность скоростей автомобиля Поликарпа может иметь вид: 5, 7, 8, 6. Таким образом, суммарный путь составит $5 + 7 + 8 + 6 = 26$ метров.

Во втором примере, так как $d = 0$, автомобиль проезжает весь участок с постоянной скоростью $v = 10$. За $t = 10$ секунд он проедет расстояние в 100 метров.



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Нечетная сумма

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Дана последовательность из n целых чисел a_1, a_2, \dots, a_n . Необходимо найти такую её подпоследовательность, что её сумма нечётна и максимальна по всем таким подпоследовательностям. Гарантируется, что в заданной последовательности найдется подпоследовательность с нечетной суммой.

Подпоследовательность — это последовательность, которую можно получить из другой последовательности путем удаления некоторых элементов, не меняя порядок оставшихся элементов.

Напишите программу, которая найдет сумму искомой подпоследовательности.

Входные данные

В первой строке задано целое число n ($1 \leq n \leq 10^5$).

Во второй строке задано n целых чисел a_1, a_2, \dots, a_n ($-10^4 \leq a_i \leq 10^4$). В данной последовательности есть хотя бы одна подпоследовательность с нечетной суммой.

Выходные данные

Выведите сумму искомой подпоследовательности.

Примеры

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

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

Примечание

В первом примере сумма второго и четвертого элементов равна 3.



C. Конфеты!

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Рассмотрим последовательность цифр длины 2^k $[a_1, a_2, \dots, a_{2^k}]$. Будем выполнять с ней следующую операцию: заменить пару (a_{2i+1}, a_{2i+2}) на $(a_{2i+1} + a_{2i+2}) \bmod 10$ для $0 \leq i < 2^{k-1}$. За каждое i такое, что $a_{2i+1} + a_{2i+2} \geq 10$, мы получаем конфету! В результате, мы получим последовательность цифр длины 2^{k-1} .

Менее формально, последовательность длины 2^k мы разбиваем на 2^{k-1} пар, каждая из которых состоит из двух чисел: первая пара состоит из первых двух чисел, вторая из третьего и четвертого, ..., последняя пара состоит из $(2^k - 1)$ -го и (2^k) -го чисел. За каждую пару, сумма чисел в которой не менее 10, мы получаем конфету. После этого, мы заменяем каждую пару чисел на остаток от деления их суммы на 10 (при этом сохраняем порядок чисел).

Будем выполнять данную операцию с последовательностью, пока она не станет длины 1. Пусть $f([a_1, a_2, \dots, a_{2^k}])$ обозначает количество конфет, которые мы получим в процессе.

К примеру: если изначальной последовательностью была $[8, 7, 3, 1, 7, 0, 9, 4]$, то:

После первой операции последовательность станет $[(8 + 7) \bmod 10, (3 + 1) \bmod 10, (7 + 0) \bmod 10, (9 + 4) \bmod 10] = [5, 4, 7, 3]$, и мы получим 2 конфеты так как $8 + 7 \geq 10$ и $9 + 4 \geq 10$.

После второй операции последовательность станет $[(5 + 4) \bmod 10, (7 + 3) \bmod 10] = [9, 0]$, и мы получим еще одну конфету так как $7 + 3 \geq 10$.

После последней операции последовательность станет $[(9 + 0) \bmod 10] = [9]$.

Следовательно, $f([8, 7, 3, 1, 7, 0, 9, 4]) = 3$, так как мы получили всего 3 конфеты.

Вам дана последовательность цифр длины n s_1, s_2, \dots, s_n . Вы должны ответить на q запросов вида (l_i, r_i) , где для i -го запроса вы должны вывести $f([s_{l_i}, s_{l_i+1}, \dots, s_{r_i}])$. Гарантируется, что $r_i - l_i + 1$ имеет вид 2^k для некоторого неотрицательного целого k .

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — длину последовательности.

Вторая строка содержит n цифр s_1, s_2, \dots, s_n ($0 \leq s_i \leq 9$).

Третья строка содержит одно целое число q ($1 \leq q \leq 10^5$) — количество запросов.

Каждая из следующих q строк содержит два целых числа l_i, r_i ($1 \leq l_i \leq r_i \leq n$) — i -й запрос. Гарантируется, что $r_i - l_i + 1$ является неотрицательной целой степенью 2.

Выходные данные

Выведите q строк, в i -й строке выведите единственное число — $f([s_{l_i}, s_{l_i+1}, \dots, s_{r_i}])$, ответ на i -й запрос.

Примеры

входные данные	Скопировать
8 8 7 3 1 7 0 9 4 3 1 8 2 5 7 7	
выходные данные	Скопировать
3 1 0	
входные данные	Скопировать
6 0 1 2 3 3 5 3 1 2	

1 4
3 6

выходные данные

Скопировать

0
0
1

Примечание

Первый пример иллюстрирует пример из условия

$f([7, 3, 1, 7]) = 1$: последовательность операций такая — $[7, 3, 1, 7] \rightarrow [(7 + 3) \bmod 10, (1 + 7) \bmod 10] = [0, 8]$ и одна конфета так как $7 + 3 \geq 10 \rightarrow [(0 + 8) \bmod 10] = [8]$, поэтому мы получаем всего 1 конфету.

$f([9]) = 0$, так как мы не выполняем с ним операции.

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 12.11.2024 20:37:37^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ЗАДАЧИ

ОТΟΣЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

C. Попрыгунчик

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Вы создаете уровень для некоторой мобильной игры. Уровень состоит из нескольких клеточек, выстроенных в ряд слева направо и пронумерованных последовательными натуральными числами, начиная с 1. Каждую клеточку вы можете оставить пустой или расположить там платформу.

Чтобы пройти уровень, игрок должен бросить мяч слева так, чтобы он сначала упал на платформу в некоторой клеточке p , отскочил от нее, затем отскочил от платформы в клеточке $(p + k)$, затем от платформы в клеточке $(p + 2k)$, и так далее от платформы в каждой k -й клеточке до тех пор, пока он не перепрыгнет правее последней клеточки. Если хотя бы одна из этих клеточек не содержит платформы, то уровень с такими значениями p и k пройти нельзя.

У вас уже есть некоторый шаблон уровня, описываемый числами $a_1, a_2, a_3, \dots, a_n$, где $a_i = 0$ означает, что в клеточке i нет платформы, а $a_i = 1$ означает, что платформа там есть. Вы хотите модифицировать этот шаблон так, чтобы уровень можно было пройти с заданными p и k . За x секунд вы можете добавить платформу в любую пустую клеточку. За y секунд вы можете полностью убрать первую клеточку, при этом количество клеточек уменьшится на один, а оставшиеся клетки пронумеруются заново, сохраняя порядок. Других изменений вы вносить не можете. Вы **не можете** уменьшить число клеток до меньше p .

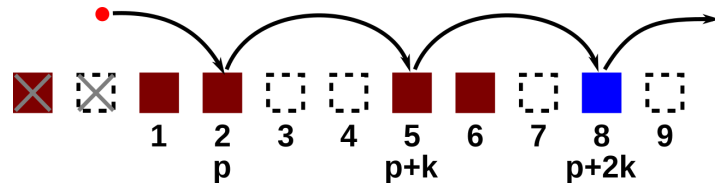


Иллюстрация к третьему тестовому случаю. Крестами отмечены удаленные клеточки. Синим показана добавленная платформа.

Какое минимальное количество секунд вам требуется, чтобы сделать возможным прохождение уровня с заданными значениями p и k ?

Входные данные
Первая строка содержит количество тестовых случаев t ($1 \leq t \leq 100$). Далее следуют описания тестовых случаев.

Первая строка каждого тестового случая содержит три целых числа n, p и k ($1 \leq p \leq n \leq 10^5, 1 \leq k \leq n$) — начальное количество клеточек, номер первой клеточки, которая должна содержать платформу, и требуемый период прыжков мяча.

Вторая строка каждого тестового случая содержит строку $a_1 a_2 a_3 \dots a_n$ ($a_i = 0$ или $a_i = 1$) — начальный шаблон уровня, записанный **без пробелов**.

Последняя строка каждого тестового случая содержит два целых числа x и y ($1 \leq x, y \leq 10^4$) — время, необходимое для добавления платформы и время, необходимое для удаления первой клеточки, соответственно.

Сумма n по всем тестовым случаям не превосходит 10^5 .

Выходные данные
Для каждого тестового случая выведите одно целое число — минимальное количество секунд, необходимое вам, чтобы изменить уровень соответствующим образом.

Можно показать, что всегда возможно изменить уровень так, чтобы его можно было пройти.

Пример	
входные данные	Скопировать
<pre>3 10 3 2 0101010101 2 2 5 4 1 00000 2 10 11 2 3 10110011000 4 3</pre>	
выходные данные	Скопировать
<pre>2 4</pre>	

10

Примечание

В первом тестовом случае лучше всего просто убрать первую клеточку, после чего все необходимые платформы будут на своих местах: ~~0~~**101010101**. Зачеркнутая цифра удалена, цифры, выделенные жирным — места, где должны быть платформы.

Необходимое время равно $y = 2$.

Во втором тестовом случае лучше всего добавить платформы в клетки 4 и 5: 000**00** → 000**11**. Необходимое время равно $x \cdot 2 = 4$.

В третьем тестовом случае лучше всего удалить первую клеточку дважды и затем добавить платформу в клеточку, которая изначально была 10-й: ~~1~~~~0~~**110011000** → ~~1~~~~0~~**110011010**. Необходимое время равно $y \cdot 2 + x = 10$.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 12.11.2024 20:37:29^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке

**ІІТМО**

С. Порванный счастливый билет

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

Билет — это непустая строка из цифр от 1 до 9.

Счастливый билет — это такой билет, у которого:

- четная длина;
- сумма цифр в первой половине равна сумме цифр во второй половине.

У вас есть n частей билетов s_1, s_2, \dots, s_n . Сколько пар (i, j) (для $1 \leq i, j \leq n$) существует таких, что $s_i + s_j$ является счастливым билетом? Обратите внимание, что возможно $i = j$.

Здесь оператор $+$ обозначает конкатенацию двух строк. Например, если s_i — это 13, а s_j — это 37, то $s_i + s_j$ будет 1337.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество частей билетов.

Вторая строка содержит n непустых строк s_1, s_2, \dots, s_n , каждая длиной не более 5 и состоящая только из цифр от 1 до 9.

Выходные данные

Выведите одно целое число — количество пар (i, j) (для $1 \leq i, j \leq n$), таких что $s_i + s_j$ является счастливым билетом.

Примеры

входные данные	Скопировать
10 5 93746 59 3746 593 746 5937 46 59374 6	
выходные данные	Скопировать
20	
входные данные	Скопировать
5 2 22 222 2222 22222	
выходные данные	Скопировать
13	
входные данные	Скопировать
3 1 1 1	
выходные данные	Скопировать
9	



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

B1. К по цене одного (простая версия)

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Это простая версия этой задачи. Единственное отличие в ограничении на k — количество подарков в акции. В этой версии, $k = 2$.

Вася пришел в магазин, чтобы купить подарки для своих друзей на Новый год. Оказалось, что ему очень повезло — именно сегодня в магазине проводится акция « k товаров по цене одного». Помните, что в этой задаче $k = 2$.

Благодаря этой акции Вася может купить ровно k любых подарков, заплатив только за самый дорогой из них. Вася решил воспользоваться этой возможностью и купить как можно больше подарков для своих друзей на деньги, которые у него имеются.

Более формально, для каждого подарка определена его цена a_i — количество монет, которое нужно потратить, чтобы приобрести этот подарок. Изначально, у Васи есть p монет. Он хочет приобрести максимальное количество подарков. Вася может сколько угодно раз выполнить одну из следующих операций.

- Вася может купить один любой подарок с номером i , если у него в настоящий момент достаточно монет (то есть $p \geq a_i$). После покупки этого подарка, количество монет у Васи уменьшится на величину a_i , то есть становится $p := p - a_i$.
- Вася может купить подарок с номером i , а также выбрать еще ровно $k - 1$ подарков, цены которых не превосходят a_i , если у него в настоящий момент достаточно монет (то есть $p \geq a_i$). Таким образом он покупает все эти k подарков, а его количество монет уменьшается на величину a_i , то есть становится $p := p - a_i$.

Обратите внимание, что каждый подарок можно приобрести не больше одного раза.

Например, если в магазине сейчас есть $n = 5$ подарков, имеющих цены $a_1 = 2, a_2 = 4, a_3 = 3, a_4 = 5, a_5 = 7$ соответственно, $k = 2$, а у Васи есть 6 монет, то он может купить суммарно 3 подарка. Подарок с номером 1 Вася купит, не используя акцию, и заплатит 2 монеты. Подарки с номерами 2 и 3 Вася купит используя акцию, и заплатит 4 монеты. Можно доказать, что приобрести больше подарков, имея шесть монет, Вася не может.

Помогите Васе узнать максимальное количество подарков, которые он может приобрести.

Входные данные

В первой строке находится число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Далее следуют описания t наборов входных данных, по две строки на каждый набор.

В первой строке каждого набора входных данных содержится три целых числа n, p, k ($2 \leq n \leq 2 \cdot 10^5, 1 \leq p \leq 2 \cdot 10^9, k = 2$) — количество товаров в магазине, количество монет, имеющихся у Васи и количество товаров, которые можно купить по цене самого дорогого.

Во второй строке каждого набора данных находятся n целых чисел a_i ($1 \leq a_i \leq 10^4$) — цены подарков.

Гарантируется, что сумма n по всем наборам данных не превосходит $2 \cdot 10^5$. Гарантируется, что в этой версии задачи $k = 2$ для всех наборов входных данных.

Выходные данные

Для каждого набора данных в отдельной строке выведите одно целое число m — максимальное количество подарков, которые может купить Вася.

Пример

входные данные	Скопировать
<pre> 6 5 6 2 2 4 3 5 7 5 11 2 2 4 3 5 7 2 10000 2 10000 10000 2 9999 2 10000 10000 5 13 2 8 2 8 2 5 3 18 2 1 2 3 </pre>	

выходные данные		Скопировать
3		
4		
2		
0		
4		
3		

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 12.11.2024 20:37:16^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

A. Просто как one и two

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

Задана непустая строка $s = s_1s_2 \dots s_n$, которая состоит исключительно из строчных букв латинского алфавита. Строка не нравится Поликарпу, если содержит в качестве **подстроки** строки хотя бы одну строку «one» или хотя бы одну строку «two» (или их обе одновременно). Иными словами, строка s не нравится Поликарпу, если существует такое целое j ($1 \leq j \leq n - 2$), что $s_js_{j+1}s_{j+2} = \text{«one»}$ или $s_js_{j+1}s_{j+2} = \text{«two»}$.

Например:

- «oneee», «ontwow», «twone» и «oneonetwo» — не нравятся Поликарпу (в них есть хотя бы одна подстрока «one» или «two»);
- «oonnee», «twwwo» и «twnoe» — нравятся Поликарпу (в них нет подстрок «one» и «two»).

Поликарп хочет выбрать некоторый набор индексов (позиций) и удалить все буквы на этих позициях. Процесс удаления букв происходит одновременно.

Например, если строка имела вид $s = \text{«onetwone»}$, то если Поликарп выберет два индекса 3 и 6, то будут выбраны «onetwone» и в результате получится «ontwne».

Какое минимальное количество индексов (позиций) надо выбрать Поликарпу, чтобы строка перестала ему не нравиться? Какие это должны быть позиции?

Входные данные

В первой строке входных данных записано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте. Далее заданы сами наборы входных данных.

Каждый набор входных данных состоит из одной непустой строки, содержащей s . Ее длина не превосходит $1.5 \cdot 10^5$. Строка s состоит исключительно из строчных букв латинского алфавита.

Гарантируется, что сумма длин всех строк по всем наборам входных данных в тесте не превосходит $1.5 \cdot 10^6$.

Выходные данные

Для каждого набора входных данных выведите ответ.

Первая строка ответа должна содержать r ($0 \leq r \leq |s|$) — искомое минимальное количество позиций, где $|s|$ — длина заданной строки. Вторая строка ответа должна содержать r различных целых чисел от — сами индексы для удаления в произвольном порядке. Индексы нумеруются слева направо от 1 до длины строки. Если $r = 0$, то вторую строку можно не выводить (а можно вывести пустой). Если ответов несколько, то выведите любой из них.

Примеры

входные данные	Скопировать
4 onetwone testme oneoneone twotwo	
выходные данные	Скопировать
2 6 3 0 3 4 1 7 2 1 4	
входные данные	Скопировать
10 onetwonetwooneoonetwooo two one twooooo	

ttttwo
ttwwoo
ooone
onppe
oneeeee
oneeeeeetwooooo

выходные данные

Скопировать

6
18 11 12 1 6 21
1
1
1
3
1
2
1
6
0

1
4
0

1
1
2
1 11

Примечание

В первом примере ответы равны:

- «onetwo»;
- «testme» — строка уже нравится Поликарпу, ничего удалять не надо;
- «oneoneone»;
- «twotwo».

Во втором примере ответы равны:

- «onetwonetwooneooonetwooo»;
- «two»;
- «one»;
- «twooooo»;
- «tttttwo»;
- «ttwwoo» — строка уже нравится Поликарпу, ничего удалять не надо;
- «ooone»;
- «onppe» — строка уже нравится Поликарпу, ничего удалять не надо;
- «oneeeee»;
- «oneeeeeetwooooo».

При поддержке



ЗАДАЧИ

ОТОСЛАТЬ

СТАТУС

ПОЛОЖЕНИЕ

ЗАПУСК

В. Найди ель

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

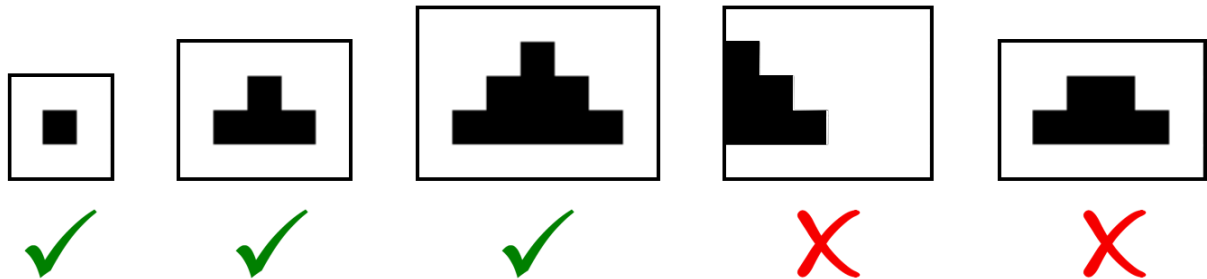
Вот-вот наступит новый год. Рик понял, что ему пора задуматься о покупке традиционных новогодних елок. Но Рик устал покупать настоящие елки, поэтому он решил найти их в матрице размером $n \times m$, состоящей из символов «*» и «.».



Для того, чтобы найти все ели, сперва дадим определение ели в матрице. Набор ячеек из матрицы называется елью с высотой k и вершиной в точке (x, y) , если:

- Все ячейки из набора являются символами «*».
- Для всех $1 \leq i \leq k$ все ячейки с номером строки $x + i - 1$ и с номерами столбцов в диапазоне $[y - i + 1, y + i - 1]$ должны принадлежать набору. Все остальные ячейки не могут принадлежать набору.

Примеры корректных и некорректных елок:



Теперь Рик стало интересно, сколько елей существует в его матрице $n \times m$. Помогите ему ответить на этот вопрос.

Входные данные

Каждый тест содержит один или несколько наборов входных данных. В первой строке записано количество наборов входных данных t ($1 \leq t \leq 10$).

Первая строка каждого набора входных данных содержит два целых числа n и m ($1 \leq n, m \leq 500$) — размер матрицы.

Следующие n строк каждого из наборов содержат по m символов $c_{i,j}$ — описание матрицы. Гарантируется, что $c_{i,j}$ является либо символом «.», либо символом «*».

Гарантируется, что сумма $n \cdot m$ по всем тестовым наборам не превосходит 500^2 ($\sum n \cdot m \leq 500^2$).

Выходные данные

Выведите единственное целое число — количество елок в матрице.

Пример

входные данные	Скопировать
4 2 3 .*. *** 2 3 .*.	

```

**
4 5
.***
*****
*****
*.*
*.*
5 7
.*.*
.***
*****
*****
.***
.*.*

```

выходные данные	Скопировать
5 3 23 34	

Примечание
В первом тестовом примере первая ель имеет вершину в точке (1, 2) и высоту 2, вторая ель имеет вершину в точке (1, 2) и высоту 1, третья ель имеет вершину в точке (2, 1) и высоту 1, четвертая ель имеет вершину в точке (2, 2) и высоту 1, пятая ель имеет вершину в точке (2, 3) и высоту 1.

Во втором тестовом примере первая ель имеет вершину в точке (1, 2) и высоту 1, вторая ель имеет вершину в точке (2, 1) и высоту 1, третья ель имеет вершину в точке (2, 2) и высоту 1.



С. Кто-нибудь хочет вампирские способности?

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

ДИО знает, что Крестоносцы звёздной пыли определили его местоположение и собираются сразиться с ним. Чтобы сорвать их планы, он решает послать несколько пользователей **стендов** для борьбы с ними. Первоначально он призвал с собой n пользователей, причем сила i -го из них равна a_i . Используя свои вампирические способности, он может делать следующее столько раз, сколько пожелает:

- Пусть **текущее** число пользователей стенда равно m .
- ДИО выбирает индекс i ($1 \leq i \leq m$).
- Затем он вызывает нового пользователя стенда с индексом $m + 1$ и силой, равной

$$a_{m+1} = a_i \oplus a_{i+1} \oplus \dots \oplus a_m,$$

где оператор \oplus обозначает операцию **побитового исключающего ИЛИ**.

- В итоге число пользователей стенда становится $m + 1$.

К несчастью для ДИО, используя предсказательные способности Пурпурного отшельника, Крестоносцы знают, что он замышляет это, и они также знают силы первоначальных пользователей стендов. Помогите Крестоносцам найти максимально возможную **силу** пользователя стенда среди всех возможных способов вызовов.

Входные данные

Каждый тест содержит несколько наборов входных данных. Первая строка содержит количество наборов входных данных t ($1 \leq t \leq 10\,000$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 10^5$) — количество пользователей Стенда, вызванных первоначально.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^8$) — сила каждого пользователя Стенда.

Гарантируется, что сумма n по всем наборам входных данных не превышает 10^5 .

Выходные данные

Для каждого набора входных данных выведите одно целое число, максимальную **силу** пользователя Стенда среди всех возможных способов вызовов.

Пример

<div>входные данные</div> <div>3 4 0 2 5 1 3 1 2 3 5 8 2 4 12 1</div>	<div>Скопировать</div>
<div>выходные данные</div> <div>7 3 14</div>	<div>Скопировать</div>

Примечание

В первом наборе входных данных один из способов добавления новых пользователей Стендов выглядит следующим образом:

- Выбрать $i = n$. Теперь a становится $[0, 2, 5, 1, 1]$.
- Выбрать $i = 1$. Теперь a становится $[0, 2, 5, 1, 1, 7]$. 7 — это максимальная **сила** пользователя стенда, которого может вызвать ДИО.

Во втором наборе входных данных, ДИО не нужно добавлять больше пользователей стенда, потому что 3 — это максимальная **сила** пользователя стенда, которого ДИО может вызвать.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Количество произведений

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

В данной задаче вам задана последовательность a_1, a_2, \dots, a_n , состоящая из n ненулевых целых чисел (то есть $a_i \neq 0$).

Перед вами стоит задача найти два числа:

- 1. количество таких пар индексов (l, r) ($l \leq r$), что произведение $a_l \cdot a_{l+1} \cdot \dots \cdot a_{r-1} \cdot a_r$ строго отрицательно;
- 2. количество таких пар индексов (l, r) ($l \leq r$), что произведение $a_l \cdot a_{l+1} \cdot \dots \cdot a_{r-1} \cdot a_r$ строго положительно.

Входные данные

В первой строке следует целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество элементов в последовательности.

Во второй строке следует n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9; a_i \neq 0$) — элементы последовательности.

Выходные данные

Выведите два целых числа — количество отрезков со строго отрицательным произведением и количество отрезков со строго положительным произведением.

Примеры

входные данные	Скопировать
5 5 -3 3 -1 1	
выходные данные	Скопировать
8 7	

входные данные	Скопировать
10 4 2 -4 3 1 2 -4 3 2 3	
выходные данные	Скопировать
28 27	

входные данные	Скопировать
5 -1 -2 -3 -4 -5	
выходные данные	Скопировать
9 6	



[ЗАДАЧИ](#) [ОТΟΣЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Игра с перестановкой

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Алиса и Боб играют в игру. У них есть перестановка p размера n (перестановка размера n — это массив размера n , где каждый элемент от 1 до n встречается ровно один раз). Также у них есть фишка, которую можно разместить на любом элементе перестановки.

Алиса и Боб делают ходы поочередно: Алиса делает первый ход, затем Боб делает второй ход, затем Алиса делает третий ход и так далее. Во время первого хода Алиса выбирает любой элемент перестановки и ставит фишку на этот элемент. Во время каждого из следующих ходов текущий игрок **должен** переместить фишку на элемент, который одновременно находится слева и строго меньше текущего элемента (т.е. если фишка находится на i -м элементе, она может быть перемещена на j -й элемент, если $j < i$ и $p_j < p_i$). Если игрок не может сделать ход (переместить фишку в соответствии с правилами игры невозможно), этот игрок **выигрывает** в игре.

Скажем, что i -й элемент перестановки **счастливый**, если выполняется следующее условие:

- если Алиса поставит фишку на i -й элемент во время своего первого хода, она может выиграть игру независимо от того, как играет Боб (т.е. у нее есть выигрышная стратегия).

Ваша задача — посчитайте количество счастливых элементов в перестановке.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество элементов в перестановке.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$). Все p_i различны.

Сумма n по всем наборам входных данных не превышает $3 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число — количество счастливых элементов в перестановке.

Пример

входные данные	Скопировать
4 3 2 1 3 2 2 1 3 1 2 3 4 2 1 4 3	
выходные данные	Скопировать
1 0 1 2	

Примечание

В первом наборе входных данных из примера 3-й элемент — счастливый.

Во втором наборе входных данных из примера нет счастливых элементов.

В третьем наборе входных данных из примера 2-й элемент — счастливый.

В четвертом наборе входных данных из примера 3-й и 4-й элемент — счастливые.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Три табло

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Середина 2018-го года и на границе Краснокаменска, что в Забайкальском крае, Мария Степановна хочет арендовать три электронных табло для обращения внимания жителей на важную проблему.

Вдоль дороги в один ряд стоят n электронных табло, причем на i -м табло текст может отображаться только со шрифтом размера s_i . Мария Степановна хочет арендовать три таких табло с индексами $i < j < k$, что размер шрифта строго увеличиваются при проезде в определенную сторону, а именно, должно выполняться $s_i < s_j < s_k$.

Стоимость аренды i -го табло равна c_i . Определите, какую минимальную стоимость должна заплатить Мария Степановна.

Входные данные

В первой строке находится одно целое число n ($3 \leq n \leq 3\,000$) — число электронных табло.

Во второй строке находятся n целых чисел s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^9$) — размеры шрифтов на табло при движении вдоль дороги.

В третьей строке находятся n целых чисел c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^8$) — стоимости аренды табло.

Выходные данные

Если не существует трех подходящих табло, выведите -1. В противном случае, выведите одно целое число — минимальную стоимость аренды трех табло с индексами $i < j < k$ таких, что $s_i < s_j < s_k$.

Примеры

входные данные	Скопировать
5 2 4 5 4 10 40 30 20 10 40	
выходные данные	Скопировать
90	
входные данные	Скопировать
3 100 101 100 2 4 5	
выходные данные	Скопировать
-1	
входные данные	Скопировать
10 1 2 3 4 5 6 7 8 9 10 10 13 11 14 15 12 13 13 18 13	
выходные данные	Скопировать
33	

Примечание

В первом примере можно, например, выбрать табло 1, 4 и 5, так как $s_1 < s_4 < s_5$ ($2 < 4 < 10$), а суммарная стоимость равна $40 + 10 + 40 = 90$.

Во втором примере невозможно выбрать подходящую тройку табло, поэтому ответ -1.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Увеличь суммы подотрезков

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Задан массив a_1, a_2, \dots, a_n , состоящий из n целых чисел. Также дано целое число x .

Пусть $f(k)$ будет равно максимальной сумме последовательного подмассива a после применения следующей операции: прибавить x к элементам на ровно k **различных** позициях. Пустой подмассив тоже рассматривается, его сумма равна 0.

Обратите внимание, что подмассив не обязан включать в себя все увеличенные элементы.

Посчитайте максимальное значение $f(k)$ для каждого k от 0 до n независимо.

Входные данные
В первой строке записано одно целое число t ($1 \leq t \leq 5000$) — количество наборов входных данных.

В первой строке каждого набора записано два целых числа n и x ($1 \leq n \leq 5000$; $0 \leq x \leq 10^5$) — количество элементов массива и значение для прибавления.

Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($-10^5 \leq a_i \leq 10^5$).

Сумма n по всем наборам целых чисел не превосходят 5000.

Выходные данные
На каждый набор входных данных выведите $n + 1$ целое число — максимальное значение $f(k)$ для каждого k от 0 до n независимо.

Пример

входные данные	Скопировать
<pre>3 4 2 4 1 3 2 3 5 -2 -7 -1 10 2 -6 -1 -2 4 -6 -1 -4 4 -5 -4</pre>	
выходные данные	Скопировать
<pre>10 12 14 16 18 0 4 4 5 4 6 6 7 7 7 8 8 8 8</pre>	

Примечание
В первом наборе не важно, к каким элементам прибавлять x . Подотрезок с максимальной суммой всегда будет целым массивом. Если увеличить k элементов на x , к сумме прибавится $k \cdot x$.

Во втором наборе:

- Для $k = 0$ пустой подмассив — это лучший вариант.
- Для $k = 1$ наиболее выгодно увеличить элемент на позиции 3. Лучшая сумма становится $-1 + 5 = 4$ для подмассива $[3, 3]$.
- Для $k = 2$ наиболее выгодно увеличить элемент на позиции 3 и любой другой элемент. Лучшая сумма останется 4 для подотрезка $[3, 3]$.
- Для $k = 3$ необходимо увеличить все элементы. Лучшая сумма становится $(-2 + 5) + (-7 + 5) + (-1 + 5) = 5$ для подотрезка $[1, 3]$.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

C. Qpwoeirut и город

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Qpwoeirut занялся архитектурой и решил перестроить свой город.

Город состоит из n зданий, расположенных в ряд. Здание i ($1 \leq i \leq n$) имеет высоту h_i этажей. Вы можете считать, что все этажи в этой задаче имеют одинаковую высоту. Иными словами, здание i выше здания j тогда и только тогда, когда количество этажей h_i в здании i больше количества этажей h_j в здании j .

Здание i *классное*, если оно выше $(i - 1)$ -го здания и $(i + 1)$ -го здания одновременно (и оба этих здания существуют). В частности, 1-е и n -е здания не могут быть классными.

Qpwoeirut хочет максимизировать количество классных зданий. Чтобы сделать это, он может строить новые этажи на уже существующих зданиях, тем самым делая их выше. Обратите внимание, что Qpwoeirut не может удалять уже существующие этажи.

Так как новые этажи очень дорогие, Qpwoeirut хочет минимизировать количество новых этажей. Найдите минимальное количество этажей, которое ему придётся построить, чтобы максимизировать количество классных зданий.

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют описания этих наборов.

В первой строке дано одно число n ($3 \leq n \leq 10^5$) — количество зданий в городе.

Во второй строке даны n целых чисел h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$) — высоты зданий в этажах.

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

Выходные данные

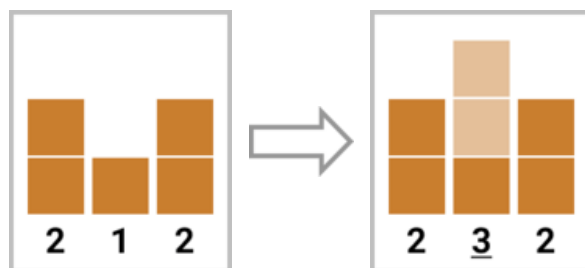
Для каждого набора входных данных выведите одно число: минимальное количество этажей, которое Qpwoeirut должен построить, чтобы максимизировать количество классных зданий.

Пример

входные данные	Скопировать
6 3 2 1 2 5 1 2 1 4 3 6 3 1 4 5 5 2 8 4 2 1 3 5 3 6 1 6 1 10 1 1 10 1 8 1 10 11 1 10 11 10 1	
выходные данные	Скопировать
2 0 3 3 0 4	

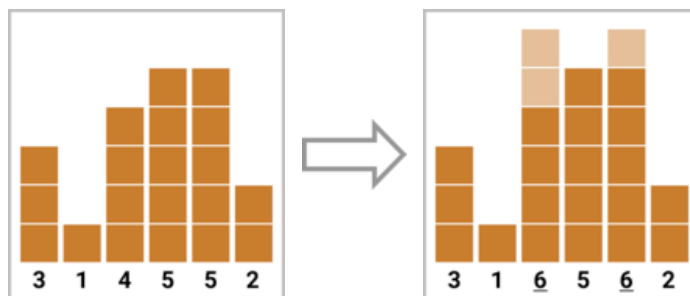
Примечание

В первом наборе входных данных оптимально построить 2 новых этажа на втором здании, в результате чего оно станет выше обоих соседних зданий. Высоты зданий будут равны $[2, \underline{3}, 2]$.



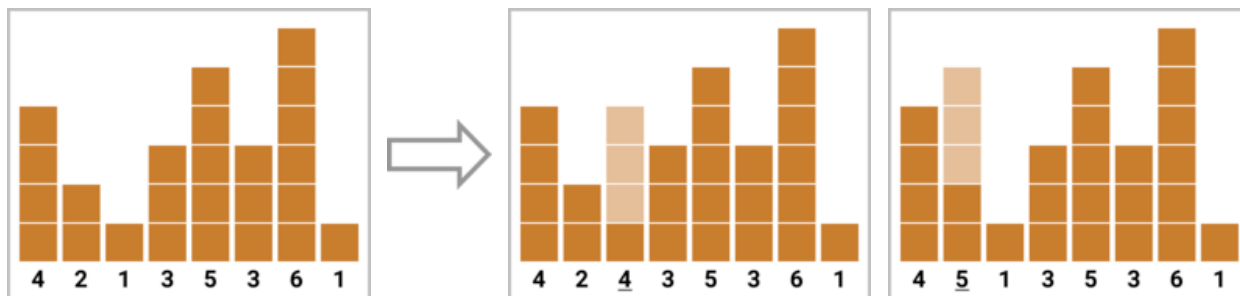
Во втором наборе входных данных количество классных зданий уже максимально, поэтому Qrwoeirut может ничего не делать.

В третьем наборе входных данных оптимально построить 2 этажа на третьем здании и 1 этаж на пятом здании. Количество классных зданий будет равно двум. Высоты зданий будут равны $[3, 1, \underline{6}, 5, \underline{6}, 2]$.



Можно показать, что нельзя сделать более 2 классных зданий или сделать 2 классных здания, построив не более 3 этажей.

В четвёртом наборе входных данных можно сделать классным второе либо третье здание. В обоих случаях надо построить 3 новых этажа. Высоты зданий будут равны $[4, 2, \underline{4}, 3, 5, 3, 6, 1]$ либо $[4, \underline{5}, 1, 3, 5, 3, 6, 1]$.



[Codeforces](https://codeforces.com/) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 12.11.2024 20:36:21^{UTC+5} (1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Разнообразная гирлянда

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

У вас есть гирлянда, состоящая из n ламп. Каждая лампа — красная, зеленая или синяя. Цвет i -й лампы равен s_i ('R', 'G' и 'B' — цвета ламп в гирлянде).

Вы хотите перекрасить некоторые лампы в этой гирлянде (перекрашивание лампы означает изменение ее изначального цвета на другой) таким образом, чтобы получившаяся гирлянда стала **разнообразной**.

Гирлянда называется **разнообразной**, если любые ее две соседние (последовательные) лампы (то есть такие лампы, что расстояние между их позициями равно 1) имеют различные цвета.

Другими словами, если получилась гирлянда t , то для всех i от 1 до $n - 1$ должно выполняться условие $t_i \neq t_{i+1}$.

Среди всех способов перекрасить изначальную гирлянду в **разнообразную** вы должны выбрать тот, в котором количество перекрашенных ламп **минимально**. Если существует несколько оптимальных решений, выведите **любое** из них.

Входные данные

Первая строка входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество ламп.

Вторая строка входных данных содержит строку s , состоящую из n символов 'R', 'G' и 'B' — цвета ламп в гирлянде.

Выходные данные

В первой строке выходных данных выведите одно целое число r — **минимальное** количество перекрашиваний, необходимых для того, чтобы получить **разнообразную** гирлянду из заданной.

Во второй строке выходных данных выведите одну строку t длины n — **разнообразную** гирлянду, полученную из изначальной за **минимальное** количество перекрашиваний. Если существует несколько оптимальных решений, выведите **любое** из них.

Примеры

входные данные	Скопировать
9 RBGRRBRGG	
выходные данные	Скопировать
2 RBGRGBRGR	
входные данные	Скопировать
8 BBBGBRRR	
выходные данные	Скопировать
2 BRBGBRGR	
входные данные	Скопировать
13 BBRRRRGGGGRR	
выходные данные	Скопировать
6 BGRBRBGBGBRG	

CODEFORCES

Sponsored by TON

Кар6502

Выйти

ЗАДАЧИОТОСЛАТЬСТАТУСПОЛОЖЕНИЕЗАПУСК

С. Равенство строк

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

У Ashish есть две строки a и b длины n и целое число k . Строки содержат только строчные буквы латинского алфавита.

Он хочет превратить строку a в строку b , исполнив несколько (возможно, ноль) операций над a .

За одну операцию он может сделать одно из двух возможных действий:

- выбрать индекс i ($1 \leq i \leq n - 1$) и поменять местами a_i и a_{i+1} , или
- выбрать индекс i ($1 \leq i \leq n - k + 1$) и, если **все** символы среди $a_i, a_{i+1}, \dots, a_{i+k-1}$ **равны** какому-то символу c ($c \neq \text{«Z»}$), заменить каждый из них на символ $(c + 1)$, таким образом, «a» заменяется на «b», «b» заменяется на «c» и так далее.

Обратите внимание, что он может исполнить любое число операций, и операции можно выполнять только на строке a .

Помогите Ashish определить, возможно ли превратить a в b , сделав несколько (возможно, ноль) операций на ней.

Входные данные

В первой строке записано одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных записаны два целых числа n ($2 \leq n \leq 10^6$) и k ($1 \leq k \leq n$).

Во второй строке записана одна строка a длины n , состоящая только из строчных букв латинского алфавита.

В третьей строке записана одна строка b длины n , состоящая только из строчных букв латинского алфавита.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6 .

Выходные данные

Для каждого набора входных данных выведите «Yes», если Ashish может превратить a в b после некоторого числа операций, иначе выведите «No».

Вы можете выводить каждый символ в любом регистре (верхнем или нижнем).

Пример

входные данные	Скопировать
4 3 3 abc bcd 4 2 abba azza 2 1 zz aa 6 2 aaabba ddddcc	
выходные данные	Скопировать
No Yes No Yes	

Примечание

В первом наборе входных данных можно доказать, что невозможно превратить a в b .

Во втором наборе входных данных

«abba» $\xrightarrow{\text{inc}}$ «acca» $\xrightarrow{\text{inc}}$... $\xrightarrow{\text{inc}}$ «azza».

Здесь «swap» обозначает операцию первого типа, а «inc» обозначает операцию второго типа.

https://codeforces.com/problemset/problem/1451/C

1/2

[ЗАДАЧИ](#) [ОТΟΣЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

А. Планирование путешествия

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Таня запланировала путешествие по городам Берляндии. Всего в Берляндии n городов, расположенных вдоль главного железнодорожного пути. Города пронумерованы от 1 до n .

План путешествия Тани выглядит следующим образом: для начала она выбирает город c_1 , с которого она начнет путешествие. Она посещает его, а затем отправляется в другой город $c_2 > c_1$, затем в следующий город $c_3 > c_2$, и так далее, пока не решит закончить свое путешествие в некотором городе $c_k > c_{k-1}$. Таким образом, последовательность посещенных городов $[c_1, c_2, \dots, c_k]$ должна быть строго возрастающей.

На последовательность посещенных городов есть еще одно ограничение. Город i имеет красоту b_i . Если Таня решит посетить только один город, красоты городов не накладывают никаких дополнительных ограничений; однако если она решила посетить несколько городов, то для каждой пары соседних городов c_i и c_{i+1} в плане должно выполняться условие $c_{i+1} - c_i = b_{c_{i+1}} - b_{c_i}$.

Например, если $n = 8$ и $b = [3, 4, 4, 6, 6, 7, 8, 9]$, следующие планы путешествия корректны:

- $c = [1, 2, 4]$;
- $c = [3, 5, 6, 8]$;
- $c = [7]$ (путешествие, состоящее из одного города, считается корректным).

Также существуют другие планы посещения городов, не описанные выше.

Таня хочет, чтобы ее путешествие было максимально красивым. Красота путешествия равна суммарной красоте всех посещенных городов. Можете ли вы составить план путешествия, максимизирующий суммарную красоту посещенных городов?

Входные данные

Первая строка содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество городов в Берляндии.

Вторая строка содержит n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 4 \cdot 10^5$), где b_i равно красоте i -го города.

Выходные данные

Выведите одно число — максимально возможную красоту путешествия Тани.

Примеры

входные данные	Скопировать
6 10 7 1 9 10 15	
выходные данные	Скопировать
26	
входные данные	Скопировать
1 400000	
выходные данные	Скопировать
400000	
входные данные	Скопировать
7 8 9 26 11 12 29 14	
выходные данные	Скопировать
55	

Примечание

Оптимальный план путешествия в первом примере: $c = [2, 4, 5]$.

Оптимальный план путешествия во втором примере: $c = [1]$.

Оптимальный план путешествия в третьем примере: $c = [3, 6]$.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Нестабильная строка

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам задана строка s , состоящая из символов 0, 1 и ?. Назовем строку **нестабильной**, если она состоит из символов 0 и 1 и любые два соседних символа различаются (т. е. имеет вид 010101... или 101010...). Назовем строку **красивой**, если она состоит из символов 0, 1 и ?, и в ней можно заменить символы ? на 0 или 1 (для каждого символа выбор происходит независимо), чтобы строка стала **нестабильной**. Например, строки 0??10, 0 и ??? являются красивыми, а строки 00 и ?1??1 — нет. Посчитайте количество красивых непрерывных подстрок строки s .

Входные данные
Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Первая и единственная строка каждого набора содержит строку s ($1 \leq |s| \leq 2 \cdot 10^5$), состоящую из символов 0, 1 и ?. Гарантируется, что сумма длин строк по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные
Для каждого набора выходных данных выведите одно целое число — количество **красивых** подстрок строки s .

Пример

входные данные	Скопировать
3 0?10 ??? ?10??1100	
выходные данные	Скопировать
8 6 25	



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Пингвин Поло и матрица

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У маленького пингвина Поло есть матрица $n \times m$, состоящая из целых чисел. Пронумеруем строки матрицы от 1 до n сверху вниз, а столбцы от 1 до m слева направо. Обозначим через a_{ij} элемент матрицы, стоящий на пересечении i -ой строки и j -го столбца.

За один шаг пингвин может добавить к любому элементу матрицы или отнять от любого элемента матрицы число d . Найдите минимальное количество шагов, которое требуется для того, чтобы все элементы матрицы были равны между собой. Если описанное невозможно, сообщите об этом.

Входные данные

В первой строке заданы три целых числа n , m и d ($1 \leq n, m \leq 100, 1 \leq d \leq 10^4$) — размеры матрицы и параметр d . В следующих n строках записана матрица: j -тое целое число в i -той строке — элемент матрицы a_{ij} ($1 \leq a_{ij} \leq 10^4$).

Выходные данные

В единственной строке выведите целое число — минимальное количество шагов, которое требуется для того, чтобы все элементы матрицы были равны между собой. Если описанное невозможно, выведите «-1» (без кавычек).

Примеры

входные данные	Скопировать
2 2 2 2 4 6 8	
выходные данные	Скопировать
4	
входные данные	Скопировать
1 2 7 6 7	
выходные данные	Скопировать
-1	



С. Письмо

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Петр уже дописал письмо своей возлюбленной Стасе, когда заметил, что оно выглядит не очень красиво. Петр сильно нервничал пока писал письмо, поэтому некоторые буквы получились строчными, а некоторые — прописными.

По мнению Петра, сообщение является *красивым*, если любая прописная буква стоит левее любой строчной. Иными словами это правило описывает строки, в которых сначала идет ноль или более прописных букв, а затем — ноль или более строчных.

Для приведения письма в красивый вид Петр может замазать некоторую букву и записать эту же букву в этом же месте в противоположном регистре (строчную букву заменить на прописную и наоборот). Петр заинтересовался вопросом — какое наименьшее число действий необходимо, чтобы привести письмо в красивый вид. Одним действием будет считать смену регистра буквы в письме. Никаких других действий Петр совершать не может.

Входные данные
В единственной строке входных данных содержится непустая строка из строчных и прописных латинских букв. Длина строки не превосходит 10^5 .

Выходные данные
Выведите единственное число — наименьшее число действий, необходимое, чтобы письмо стало *красивым*.

Примеры

входные данные	Скопировать
PRuvetSTAAyA	
выходные данные	Скопировать
5	

входные данные	Скопировать
OYPROSTIYAOPЕCHATALSYAPRIVETSTASYA	
выходные данные	Скопировать
0	

входные данные	Скопировать
helloworld	
выходные данные	Скопировать
0	



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Аппарат Констанции

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Констанция — самая умная девушка в своей деревне, но вот зрением она не блещет.

Однажды она изобрела невероятный аппарат! Когда вы произносите букву, аппарат записывает ее на лист бумаги. Например, если вы произнесете буквы 'c', 'o', 'd' и 'e' в таком порядке, то аппарат запишет «code» на бумаге. Благодаря этому изобретению, она наконец-то может писать письма, не надевая очки.

Однако ее глупый друг Акко решил над ней подшутить. Акко испортил аппарат так, что если произнести 'w', то будет записано «ии» вместо «w», и если произнести 'm', то будет записано «пп» вместо «m». Так как зрение у Констанции довольно плохое, она не заметила, что Акко сделал.

Остальные буквы работают так же, как и прежде — если произнести любую букву, кроме 'w' и 'm', то аппарат просто запишет ее на бумаге.

На следующий день я получил письмо. Я его не смог разобрать, так что я решил, что это либо какая-то тарабарщина от Акко, либо то, что записала Констанция своим аппаратом. Но так как мне известно, что совершил Акко, мне захотелось перечислить все возможные строки, которые могли получиться в результате записи аппаратом Констанции, чтобы посмотреть, могло ли это быть посланным сообщением.

Таким образом, мне хочется узнать, сколько понадобится бумаги для этой затеи. В этом мне не помешает твоя помощь. Подскажи мне количество строк, которые аппарат Констанции мог превратить в полученное мной сообщение.

Так как данное число могло получиться довольно большим, сообщи мне остаток от его деления на $10^9 + 7$.

Если не существует ни одной такой строки, что аппарат Констанции мог превратить их в полученное мной сообщение, то выведите 0.

Входные данные
В единственной строке содержится строка s ($1 \leq |s| \leq 10^5$) — полученное сообщение. s содержит только строчные латинские буквы.

Выходные данные
Выведите одно целое число — количество строк, которые аппарат Констанции мог превратить в полученное мной сообщение s , по модулю $10^9 + 7$.

Примеры	
<div>входные данные<div>оuuоkarinn</div></div>	<div>Скопировать</div>
<div>выходные данные<div>4</div></div>	<div>Скопировать</div>
<div>входные данные<div>banana</div></div>	<div>Скопировать</div>
<div>выходные данные<div>1</div></div>	<div>Скопировать</div>
<div>входные данные<div>nnn</div></div>	<div>Скопировать</div>
<div>выходные данные<div>3</div></div>	<div>Скопировать</div>
<div>входные данные<div>amanda</div></div>	<div>Скопировать</div>
<div>выходные данные<div>0</div></div>	<div>Скопировать</div>

Примечание

В первом примере возможные строки: «ouuokari~~nn~~», «ouuokari~~m~~», «owokari~~m~~» и «owokari~~nn~~».

Во втором примере есть только одна строка: «banana».

В третьем примере возможные строки: «nm», «mn» и «nnn».

В последнем примере нет ни одной строки, которую аппарат бы вывел как «amanda», так как аппарат не может напечатать 'm'.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 12.11.2024 20:35:17^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке

**ІТМО**