


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

E. Передача последовательности по сети

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Последовательность a передается по сети следующим образом:

1. последовательность a разбивают на отрезки (каждый элемент последовательности принадлежит ровно одному отрезку, каждый отрезок — это подряд идущие элементы последовательности);
2. рядом с каждым отрезком, либо слева, либо справа от него, записывают длину этого отрезка;
3. полученная последовательность b передаётся по сети.

Например, нужно было передать последовательность $a = [1, 2, 3, 1, 2, 3]$. Допустим, что её разбили на отрезки следующим образом: $[1] + [2, 3, 1] + [2, 3]$. Тогда могли получиться следующие последовательности:

- $b = [1, \textcolor{red}{1}, 3, \textcolor{blue}{2}, \textcolor{red}{3}, 1, \textcolor{green}{2}, \textcolor{blue}{3}, 2]$,
- $b = [\textcolor{red}{1}, 1, 3, \textcolor{blue}{2}, \textcolor{red}{3}, 1, 2, \textcolor{blue}{2}, \textcolor{red}{3}]$,
- $b = [\textcolor{blue}{1}, 1, \textcolor{red}{2}, \textcolor{blue}{3}, \textcolor{red}{1}, 3, 2, \textcolor{blue}{2}, \textcolor{red}{3}]$,
- $b = [\textcolor{red}{1}, 1, \textcolor{blue}{2}, \textcolor{red}{3}, \textcolor{blue}{1}, 3, \textcolor{green}{2}, \textcolor{blue}{3}, 2]$.

Если бы было использовано другое разбиение на отрезки, то переданная последовательность могла бы быть другой.

Задана последовательность b . Могла ли последовательность b быть передана по сети? Другими словами, существует ли такая последовательность a , что при преобразовании a для передачи по сети могла получиться последовательность b ?

Входные данные

В первой строке входных данных дано единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Каждый набор входных данных состоит из двух строк.

В первой строке входных данных дано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — размер последовательности b .

Во второй строке входных данных дано n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^9$) — сама последовательность b .

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

Выходные данные

Для каждого набора входных данных в отдельной строке выведите:

- YES, если последовательность b могла быть передана по сети, то есть если последовательность b могла быть получена из некоторой последовательности a для передачи a по сети.
- NO в противном случае.

Вы можете выводить YES и NO в любом регистре (например, строки yEs, yes, Yes и YES будут распознаны как положительный ответ).

Пример

входные данные	Скопировать
7 9 1 1 2 3 1 3 2 2 3 5 12 1 2 7 5 6 5 7 8 9 10 3 4 4 8 6 2 2 3 1 10 4 6 2 1 9 4 9 3 4 2 1 1	
выходные данные	Скопировать
YES YES	

YES
NO
YES
YES
NO

Примечание

В первом наборе входных данных последовательность b могла быть получена из последовательности $a = [1, 2, 3, 1, 2, 3]$ со следующим разбиением: [1] + [2, 3, 1] + [2, 3]. Последовательность b : [1, 1, 2, 3, 1, 3, 2, 2, 3].

Во втором наборе входных данных последовательность b могла быть получена из последовательности $a = [12, 7, 5]$ со следующим разбиением: [12] + [7, 5]. Последовательность b : [12, 1, 2, 7, 5].

В третьем наборе входных данных последовательность b могла быть получена из последовательности $a = [7, 8, 9, 10, 3]$ со следующим разбиением: [7, 8, 9, 10, 3]. Последовательность b : [5, 7, 8, 9, 10, 3].

В четвертом наборе входных данных не существует такой последовательности a , что при изменении a для передачи по сети могла получиться последовательность b .

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:49:01_{UTC+5} (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



| ИТМО

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

A. DZY любит последовательности

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

У DZY есть последовательность a , состоящая из n целых чисел.

Назовем последовательность a_i, a_{i+1}, \dots, a_j ($1 \leq i \leq j \leq n$) подотрезком последовательности a . Значение $(j - i + 1)$ обозначает длину подотрезка.

DZY хочет найти максимальный по длине подотрезок a , обладающий следующим свойством. Из подотрезка можно получить строго возрастающую последовательность, выполнив изменение не более одного элемента подотрезка (разрешается изменить значение любого элемента подотрезка на любое целое значение).

Ваша задача — вывести длину искомого оптимального подотрезка.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^5$). В следующей строке записано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Выходные данные

В единственной строке выведите ответ на задачу — максимальную длину подотрезка.

Примеры

входные данные	Скопировать
6 7 2 3 1 5 6	
выходные данные	Скопировать
5	

Примечание

Вы можете выбрать подотрезок a_2, a_3, a_4, a_5, a_6 и заменить его 3-й элемент (то есть, a_4) на 4.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.11.2024 21:49:12 UTC+5 (j1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО

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

D. Скажите палиндромам - нет!

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Назовем строку **красивой**, если в ней нет подстроки длины хотя бы 2, которая является палиндромом. Напомним, что палиндром — это строка, читающаяся одинаково от первого символа к последнему и от последнего символа к первому. Например, строки a, bab, асса, bcabcbacb являются палиндромами, а строки ab, abbbaa, cccb — нет.

Определим **стоимость** строки, как минимальное количество операций, чтобы строка стала красивой, если за одну операцию разрешено изменить любой символ строки на одну из первых 3 букв латинского алфавита (в нижнем регистре).

Вам задана строка s длины n , каждый символ строки — одна из первых 3 букв латинского алфавита (в нижнем регистре).

Вам предстоит ответить на m запросов — вычислите стоимость подстроки строки s с l_i -й по r_i -ю позицию включительно.

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — длина строки s и количество запросов.

Вторая строка содержит строку s , она состоит из n символов, каждый из которых является одной из первых 3 латинских букв.

Следующие m строк содержат два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — параметры i -го запроса.

Выходные данные

Для каждого запроса выведите одно целое число — стоимость подстроки строки s с l_i -й по r_i -ю позицию включительно.

Пример

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

Примечание

Рассмотрим запросы в примере из условия.

- в первом запросе интересующая нас подстрока — baa, ее можно превратить в bac за одну операцию;
- во втором запросе интересующая нас подстрока — baacb, ее можно превратить в cbacb за две операции;
- в третьем запросе интересующая нас подстрока — cb, ее можно оставить без изменений;
- в четвертом запросе интересующая нас подстрока — aa, ее можно превратить в ba за одну операцию.



D. Последовательность и обмены

ограничение по времени на тест: 1.5 секунд

ограничение по памяти на тест: 512 мегабайт

Вам задана последовательность a , состоящая из n целых чисел a_1, a_2, \dots, a_n , а также целое число x . Ваша задача заключается в том, чтобы сделать последовательность a отсортированной (последовательность отсортирована, если выполняется условие $a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n$).

Чтобы отсортировать последовательность, вы можете выполнять следующую операцию любое количество раз (возможно, ноль): выбрать целое число i такое, что $1 \leq i \leq n$ и $a_i > x$, поменять местами значения a_i и x .

Например, если $a = [0, 2, 3, 5, 4]$, $x = 1$, возможна следующая последовательность операций:

1. выбрать $i = 2$ (это возможно, так как $a_2 > x$), получим $a = [0, 1, 3, 5, 4]$, $x = 2$;
2. выбрать $i = 3$ (это возможно, так как $a_3 > x$), получим $a = [0, 1, 2, 5, 4]$, $x = 3$;
3. выбрать $i = 4$ (это возможно, так как $a_4 > x$), получим $a = [0, 1, 2, 3, 4]$, $x = 5$.

Вычислите минимальное количество операций, которые необходимо выполнить, чтобы a стала отсортированной, или сообщите, что это невозможно.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 500$) — количество наборов входных данных.

Каждый набор входных данных состоит из двух строк. Первая строка содержит два целых числа n и x ($1 \leq n \leq 500$, $0 \leq x \leq 500$) — количество элементов в последовательности и начальное значение x .

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 500$).

Сумма значений n по всем наборам входных данных не превосходит 500.

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество операций, которые необходимо выполнить, чтобы сделать a отсортированной, или -1 , если это невозможно.

Пример

входные данные	Скопировать
<pre>6 4 1 2 3 5 4 5 6 1 1 3 4 4 1 10 2 2 10 11 9 2 10 12 11 5 18 81 324 218 413 324</pre>	
выходные данные	Скопировать
<pre>3 0 0 -1 1 3</pre>	


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

C. Голодные игры

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Ярослав играет в компьютерную игру, и на одном из уровней перед ним оказалось n выстроенных в ряд грибов. Каждый гриб имеет свой уровень ядовитости, i -й с начала гриб имеет уровень ядовитости a_i . Ярослав может выбрать два целых числа $1 \leq l \leq r \leq n$, а затем его персонаж будет по очереди слева направо поедать грибы из этого подотрезка, то есть грибы с номерами $l, l + 1, l + 2, \dots, r$.

У персонажа есть уровень отравленности g , изначально равный 0. Компьютерная игра определяется числом x — максимальным уровнем отравленности в любой момент времени. При поедании гриба с ядовитостью k происходит следующее:

1. К отравленности персонажа добавляется k .
2. Если $g \leq x$, процесс продолжается, иначе g становится равным нулю и процесс продолжается.

Ярославу стало интересно, сколько существует способов выбрать значения l и r таких, что итоговое значение g не равно нулю. Помогите Ярославу найти это число!

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит 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 два целых числа n, x ($1 \leq n \leq 2 \cdot 10^5, 1 \leq x \leq 10^9$) — количество грибов и максимальный уровень отравленности.

Вторая строка каждого набора входных данных содержит n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

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 — количество подотрезков, таких что итоговое значение g будет не равно нулю.

Пример

входные данные	Скопировать
<pre>5 4 2 1 1 1 1 3 2 1 2 3 1 6 10 6 3 1 2 1 4 3 8 5 999999999 999999999 999999998 1000000000 1000000000 500000000</pre>	
выходные данные	Скопировать
<pre>8 2 0 10 7</pre>	

Примечание

В первом наборе входных данных подходят подотрезки $(1, 1), (1, 2), (1, 4), (2, 2), (2, 3), (3, 3), (3, 4)$ и $(4, 4)$.

Во втором наборе входных данных ненулевое g останется только на подотрезках $(1, 1)$ и $(2, 2)$.

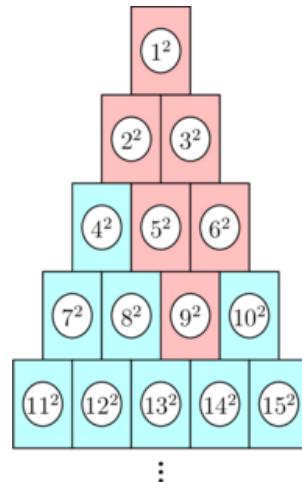
В третьем наборе входных данных на единственном возможном подотрезке g будет равно нулю.

G. Пирамида из банок

ограничение по времени на тест: 2.5 секунд

ограничение по памяти на тест: 256 мегабайт

На ярмарке есть огромная пирамида из консервных банок с 2023 рядами, пронумерованными в стандартном порядке, как показано на рисунке.



Если вначале попасть в банку с номером 9^2 , то упадут все банки, покрашенные в красный цвет на рисунке выше.

Вы бросаете мяч в пирамиду, и он попадает в одну банку с номером n^2 . Это приводит к тому, что все банки, которые находятся над этой банкой, падают (то есть банка n^2 падает, затем падают банки, непосредственно находящиеся над n^2 , затем банки, непосредственно находящиеся над этими банками, и так далее). Например, на рисунке выше показаны банки, которые упадут, если попасть в банку 9^2 .

Какова **сумма** номеров всех банок, которые упадут? Напомним, что $n^2 = n \times n$.

Входные данные

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

Единственная строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 10^6$) — это означает, что банка, в которую вы попали, имеет номер n^2 .

Выходные данные

Для каждого набора входных данных выведите одно целое число — сумму номеров всех банок, которые упадут.

Пример

входные данные	Скопировать
10 9 1 2 3 4 5 6 10 1434 1000000	
выходные данные	Скопировать
156 1 5 10 21 39 46 146 63145186 58116199242129511	

Примечание

Первый набор входных данных изображен в условии. Сумма номеров, которые упадут, равна

$$1^2 + 2^2 + 3^2 + 5^2 + 6^2 + 9^2 = 1 + 4 + 9 + 25 + 36 + 81 = 156.$$

Во втором наборе входных данных упадет только банка с номером 1^2 , поэтому ответ равен $1^2 = 1$.

В третьем наборе входных данных упадут банки с номерами 1^2 и 2^2 , поэтому ответ равен $1^2 + 2^2 = 1 + 4 = 5$.

В четвертом наборе входных данных упадут банки с номерами 1^2 и 3^2 , поэтому ответ равен $1^2 + 3^2 = 1 + 9 = 10$.

В пятом наборе входных данных упадут банки с номерами 1^2 , 2^2 и 4^2 , поэтому ответ равен $1^2 + 2^2 + 4^2 = 1 + 4 + 16 = 21$.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:49:55^{UTC+5} (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке




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

B2. К по цене одного (усложнённая версия)

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Это сложная версия этой задачи. Единственное отличие в ограничении на k — количество подарков в акции. В этой версии $2 \leq k \leq n$.

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».

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 ($2 \leq n \leq 2 \cdot 10^5, 1 \leq p \leq 2 \cdot 10^9, 2 \leq k \leq n$) — 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 $2 \cdot 10^5$.

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>8 5 6 2 2 4 3 5 7 5 11 2 2 4 3 5 7 3 2 3 4 2 6 5 2 3 10 1 3 9 2 2 10000 2 10000 10000 2 9999 2 10000 10000 4 6 4 3 2 3 2</pre>	

5	5	3		
1	2	2	1	2

ВЫХОДНЫЕ ДАННЫЕ**Скопировать**

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

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:50:01^{UTC+5} (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



| **ИТМО**


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

C. Отражения от плоскостей

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Гауранг вырос в таинственной вселенной. Перед ним стоят n 2D плоскостей одна за другой. Он пускает частицу с периодом распада k в направлении плоскостей.

Частица может пройти через плоскость напрямую, однако, каждая плоскость создает идентичную копию этой частицы, направленную в противоположном направлении с периодом распада равным $k - 1$. Если период распада частицы равен 1, то она НЕ создаст копию.

Например, если есть две плоскости, и выпущена частица с периодом распада 3 (направлена вправо), то происходит следующий процесс: (здесь $D(x)$ означает одну частицу с периодом распада x)

1. первая плоскость создает $D(2)$ влево и пропускает $D(3)$ дальше вправо;
2. вторая плоскость создает $D(2)$ влево и пропускает $D(3)$ дальше вправо;
3. первая плоскость пропускает $D(2)$ дальше влево и создает $D(1)$ вправо;
4. вторая плоскость пропускает $D(1)$ дальше вправо ($D(1)$ не может создавать копии).

В конце мультимножество S всех частиц — это $\{D(3), D(2), D(2), D(1)\}$. (Смотрите примеры для визуального отображения этого теста.)

Гауранг не справляется со сложностью этого процесса, когда количество плоскостей слишком велико. Помогите Гаурангу узнать размер мультимножества S по заданным n и k .

Так как размер мультимножества может быть очень большим, выведите его по модулю $10^9 + 7$.

Обратите внимание, что частицы могут перемещаться между плоскостями, не сталкиваясь друг с другом.

Входные данные

В первой строке записано количество наборов входных данных t ($1 \leq t \leq 100$). Затем следуют t строк, в каждой записаны два целых числа n и k ($1 \leq n, k \leq 1000$).

Сумма n по всем наборам входных данных не превосходит 1000. Сумма k по всем наборам входных данных не превосходит 1000. Все наборы входных данных в одном teste различны.

Выходные данные

Выведите t целых чисел. i -е число должно являться ответом на i -й набор входных данных.

Примеры

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

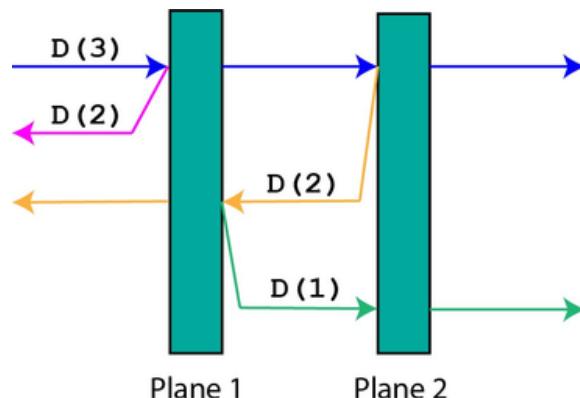
входные данные	Скопировать
3 1 1 1 500 500 250	
выходные данные	Скопировать
1 2 257950823	

Примечание

Здесь следует объяснение первого примера, состоящего из четырех наборов входных данных.

Набор входных данных 1: ($n = 2, k = 3$) уже объяснен в условии задачи.

На картинке ниже изображена симуляция. Каждая цветная прямая показывает путь одной из частиц. Как можно видеть, всего есть четыре различных частицы в мультимножестве. Обратите внимание, что вертикальные промежутки между отраженными частицами введены только для визуальной ясности (как ранее описывалось, частицы не сталкиваются друг с другом)



Набор входных данных 2: ($n = 2, k = 2$) выглядит так:

1. первая плоскость создает $D(1)$ влево и пропускает $D(2)$ дальше вправо;
2. вторая плоскость создает $D(1)$ влево и пропускает $D(2)$ дальше вправо;
3. первая плоскость пропускает $D(1)$ дальше влево ($D(1)$ не может создавать копии).

Итоговый размер мультимножества $\{D(1), D(1), D(2)\}$ равен трем.

Набор входных данных 3: ($n = 3, k = 1$) — здесь есть три плоскости, но период распада всего один. Поэтому новые копии не создаются, когда одна частица пролетает через плоскости. Поэтому ответ один.

Набор входных данных 4: ($n = 1, k = 3$) — здесь только одна плоскость. Частица создает новую копию влево.

Мультимножество $\{D(2), D(3)\}$ имеет размер два.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.11.2024 21:50:04 UTC+5 (j1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#).

При поддержке




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

C. Как ходит ладья?

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам дана шахматная доска размером $n \times n$, на которой вы и компьютер поочередно расставляете белые и черные ладьи соответственно. Расставляя ладьи, вы должны следить за тем, чтобы никакие две ладьи не атаковали друг друга. Две ладьи атакуют друг друга, если они находятся в одной строке или столбце **независимо от цвета**.

Правильный ход — установка ладьи на позиции (r, c) так, чтобы она не атаковала никакую другую ладью.

Вы начинаете первым, и когда вы в свой ход сделаете правильный ход, поставив белую ладью на позицию (r, c) , компьютер отзеркалит ваш ход и в свой ход поставит черную ладью на позицию (c, r) . Если $r = c$, то компьютер не сможет отзеркалить ваш ход и пропустит его.

Вы уже сыграли с компьютером k ходов (компьютер отразил и эти ходы), и вы должны продолжать игру, пока не останется ни одного правильного хода. Сколько различных конечных конфигураций можно получить, если продолжить игру после данных k ходов? Гарантируется, что k сделанных ходов были правильными. Так как ответ может быть большим, выведите его по модулю $10^9 + 7$.

Две конфигурации считаются различными, если существует координата (r, c) , в которой ладья есть в одной конфигурации, но нет в другой, **или** цвет ладьи на координате отличается.

Входные данные

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 два целых числа n и k ($1 \leq n \leq 3 \cdot 10^5$, $0 \leq k \leq n$) — размер шахматной доски и количество ходов, которое вы уже сыграли соответственно.

Каждая из следующих k строк набора входных данных содержит по два целых числа r_i и c_i , описывающих ваш i -й ход.

Гарантируется, что все k ходов являются правильными.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите в отдельной строке одно число — общее количество возможных конечных конфигураций по модулю $10^9 + 7$.

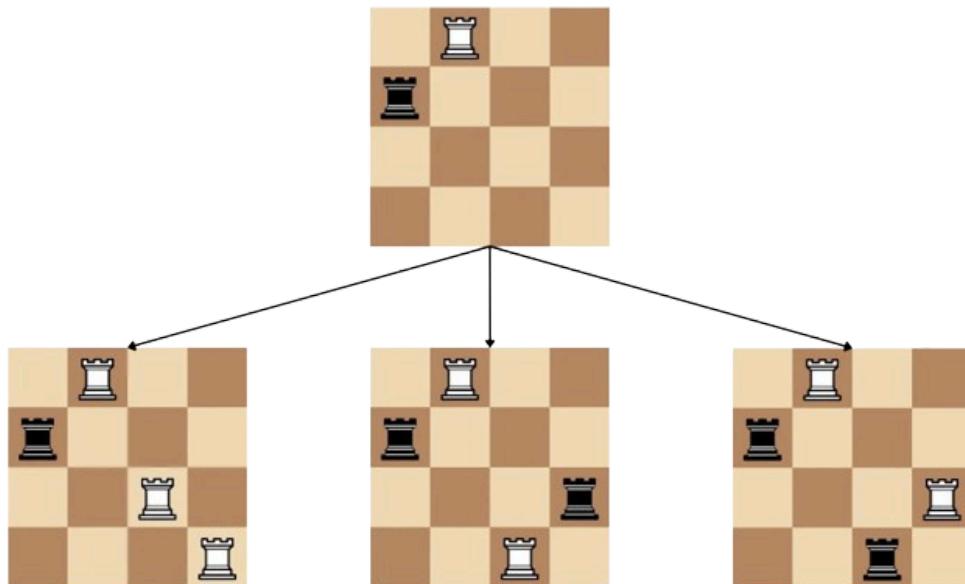
Пример

входные данные	Скопировать
<pre>3 4 1 1 2 8 1 7 6 1000 4 4 4 952 343 222 333 90 91</pre>	
выходные данные	Скопировать
<pre>3 331 671968183</pre>	

Примечание

В первом наборе входных данных у нас есть поле 4×4 , и вы уже сыграли 1 ход. После того как вы и компьютер сыграли по одному ходу, на поле есть белая ладья на $(1, 2)$ и черная ладья на $(2, 1)$. Из этого состояния возможны три конфигурации —

1. Вы ставите белую ладью на $(3, 4)$, а компьютер в ответ ставит черную ладью на $(4, 3)$.
2. Вы ставите белую ладью на $(4, 3)$, а компьютер в ответ ставит черную ладью на $(3, 4)$.
3. Вы ставите белую ладью на $(3, 3)$, а затем на $(4, 4)$, или наоборот. Оба варианта приводят к одной и той же конфигурации.



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:50:10^{UTC+5} (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке




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

A. Цепная реакция

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На различных позициях вдоль числовой прямой расположены n маяков. Координата i -го маяка равняется a_i , а уровень мощности — b_i . Когда i -й маяк активируется, он уничтожает все маяки слева от него (направление уменьшения координат) на расстоянии b_i включительно. Сам маяк при этом не уничтожается. Сайтама последовательно активирует все маяки друг за другом справа налево, при этом если маяк разрушен, то его активировать нельзя.

Сайтама хочет, чтобы Генос добавил один маяк любой мощности на любую позицию **строго правее** всех существующих маяков. При этом необходимо, чтобы в результате процесса последовательной активации уничтожалось как можно меньше маяков. Обратите внимание, что новый маяк будет расположен так, что обязательно будет активирован первым. Помогите Геносу: определите, какое минимальное количество маяков может быть уничтожено.

Входные данные

В первой строке входных данных записано единственное число n ($1 \leq n \leq 100\,000$) — количество маяков.

В каждой из следующих n строк записано по два целых числа a_i и b_i ($0 \leq a_i \leq 1\,000\,000$, $1 \leq b_i \leq 1\,000\,000$) — координата и уровень мощности соответствующего маяка. Никакие два маяка не будут расположены в одной точке, то есть $a_i \neq a_j$ if $i \neq j$.

Выходные данные

Выведите единственное число — минимально возможное количество маяков, которые могут быть уничтожены.

Примеры

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

Примечание

В первом примере минимальное количество уничтожаемых маяков равняется 1. Один из способов получить такой ответ — поместить в позицию 9 маяк с уровнем мощности 2.

Во втором примере минимальное количество уничтожаемых маяков равно 3. Одним из способов получить такой ответ является поместить в позицию 1337 маяк с уровнем мощности 42.

A. Снова функционалка

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

В Ужляндии опять что-то произошло... На улицах беспорядки... Для спасения ситуации было решено призвать знаменитых ужляндских героев — баранчика Шона и жирафика Стаса. По прибытию они обнаружили, что граждане обеспокоены незнанием крайних значений Главной Ужляндской Функции f , которая определяется следующим образом:

$$f(l, r) = \sum_{i=l}^{r-1} |a[i] - a[i + 1]| \cdot (-1)^{i-l}$$

В этой формуле $1 \leq l < r \leq n$, где n — размер Главного Ужляндского Массива a , а $|x|$ обозначает модуль числа x . Но герои прогуливали уроки математики, и поэтому они просят вашей помощи. Помогите им вычислить максимальное значение f среди всех допустимых значений l и r для заданного массива a .

Входные данные

В первой строке входного файла единственное число n ($2 \leq n \leq 10^5$) — размер массива a .

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

Выходные данные

Выведите одно целое число — максимальное значение функции f .

Примеры

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

Примечание

В первом примере максимальное значение f достигается на отрезках $[1, 2]$ и $[2, 5]$.

Во втором примере максимальное значение f достигается, только если отрезком является весь массив.




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

D. Медуза и Mex

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Вам дан массив из n целых неотрицательных чисел a_1, a_2, \dots, a_n .

Пусть m — переменная, изначально равная 0. Медуза выполняет следующую операцию n раз:

- выбрать индекс i ($1 \leq i \leq |a|$) и удалить a_i из a .
- прибавить $\text{MEX}(a)^\dagger$ к m .

Медуза хочет узнать минимально возможное конечное значение m при оптимальном выполнении всех операций.

\dagger MEX (minimum excluded) массива — это наименьшее целое неотрицательное число, которое не принадлежит массиву.

Например:

- MEX массива $[2, 2, 1]$ равен 0, так как 0 не принадлежит массиву.
- MEX массива $[3, 1, 0, 1]$ равен 2, так как 0 и 1 принадлежат массиву, а 2 - нет.
- MEX массива $[0, 3, 1, 2]$ равен 4, так как 0, 1, 2 и 3 принадлежат массиву, а 4 - нет.

Входные данные

Каждый тест содержит несколько наборов входных данных. В первой строке указывается количество наборов входных данных t ($1 \leq t \leq 5000$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 5000$) — размер массива.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — элементы массива.

Гарантируется, что сумма n по всем наборам входных данных не превышает 5000.

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное значение m при оптимальном выполнении операций.

Пример

входные данные	Скопировать
4 8 5 2 1 0 3 0 4 0 2 1 2 5 1 0 2 114514 0 8 0 1 2 0 1 2 0 3	
выходные данные	Скопировать
3 0 2 7	

Примечание

В первом наборе исходных данных мы удаляем элементы из a в следующем порядке:

$[5, 2, 1, 0, 3, 0, 4, 0] \rightarrow [5, 2, 0, 3, 0, 4, 0] \rightarrow [5, 2, 0, 3, 4, 0] \rightarrow [5, 2, 3, 4, 0] \rightarrow [5, 2, 3, 4] \rightarrow [5, 2, 4] \rightarrow [2, 4] \rightarrow [4] \rightarrow []$.

Значение m будет равно $1 + 1 + 1 + 0 + 0 + 0 + 0 + 0 = 3$.


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

B. Тренировка

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Приближается лето! Пора Яхубу и Яхубине заняться спортом, ведь они оба хотят выглядеть привлекательно на пляже. Тренажерный зал, куда они идут, представляет собой матрицу a с n строками и m столбцами. Число $a[i][j]$ обозначает калории, сжигаемые тренировкой в части спортзала, которая находится в i -й строке и j -м столбце матрицы.

Яхуб начинает тренировку в строке 1 и столбце 1. Завершить тренировку он хочет в $a[n][m]$. После завершения тренировки в $a[i][j]$, Яхуб может перейти в часть зала $a[i+1][j]$ или $a[i][j+1]$. Аналогично, Яхубина начинает тренировку в $a[n][1]$ и должна закончить тренировку в $a[1][m]$. Когда она завершает тренировку в $a[i][j]$, она идет либо в $a[i][j+1]$, либо в $a[i-1][j]$.

Тренировка Яхуба и Яхубины дополнительно должна удовлетворять еще одному условию. Им нужно встретиться ровно в одной клетке спортзала. В этой клетке никто из них не будет заниматься спортом, там они будут разговаривать об алгоритмах (передышка от тренажеров). Затем каждый из них перейдет к следующему тренажеру.

Если либо Яхуб, либо Яхубина занимался/занималась в $a[i][j]$, то число $a[i][j]$ прибавляется к общей выгоде тренировки (изначально она равна 0). Пожалуйста, составьте план тренировки для Яхуба и Яхубины, такой, чтобы общая выгода в итоге была максимальна. Обратите внимание, что Яхуб и Яхубина могут тренироваться с различными скоростями, поэтому количество пройденных частей зала до места встречи у них может различаться.

Входные данные

В первой строке записано два целых числа, n и m ($3 \leq n, m \leq 1000$). В каждой из следующих n строк записано по m целых чисел: j -е число в i -й строке обозначает $a[i][j]$ ($0 \leq a[i][j] \leq 10^5$).

Выходные данные

Выведите единственное целое число — максимальную выгоду.

Примеры

входные данные	Скопировать
3 3 100 100 100 100 1 100 100 100 100	
выходные данные	Скопировать
800	

Примечание

Яхуб выберет тренажеры $a[1][1] \rightarrow a[1][2] \rightarrow a[2][2] \rightarrow a[3][2] \rightarrow a[3][3]$. Яхубина выберет тренажеры $a[3][1] \rightarrow a[2][1] \rightarrow a[2][2] \rightarrow a[2][3] \rightarrow a[1][3]$.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
 Соревнования по программированию 2.0
 Время на сервере: 29.11.2024 21:50:21 UTC+5 (j1).
 Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО


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

A. Саша и немного отдыха

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Саша любит программировать. Однажды, во время одного очень долгого конкеста Саша решил, что он устал и было бы не плохо немного передохнуть. Так он и поступил. Но так как Саша не простой парень, то и отдыхает он необычно. Во время отдыха Саша очень любит дорешивать когда-то нерешённые задачи, ведь дорешка очень важна.

А дорешать Саша решил следующую задачу:

Дан массив a из n целых чисел, необходимо найти количество смешных пар (l, r) ($l \leq r$). Чтобы проверить, что пара (l, r) — смешная, обозначим $mid = \frac{l+r-1}{2}$, тогда, если $r - l + 1$ — четное число, и $a_l \oplus a_{l+1} \oplus \dots \oplus a_{mid} = a_{mid+1} \oplus a_{mid+2} \oplus \dots \oplus a_r$, то пара смешная. Иначе говоря \oplus элементов левой половины подмассива с l по r должен быть равен \oplus элементов правой половины. Где \oplus обозначает операцию побитового исключающего ИЛИ.

Саше необходимо продолжить решать конкест, а эту задачу он предложил решить вам.

Входные данные

Первая строка содержит одно целое число n ($2 \leq n \leq 3 \cdot 10^5$) — размер массива.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{20}$) — сам массив.

Выходные данные

Выведите одно число — количество смешных пар. Вы должны учитывать только те пары, в которых $r - l + 1$ чётно.

Примеры

входные данные	Скопировать
5 1 2 3 4 5	
выходные данные	Скопировать
1	
входные данные	Скопировать
6 3 2 2 3 7 6	
выходные данные	Скопировать
3	
входные данные	Скопировать
3 42 4 2	
выходные данные	Скопировать
0	

Примечание

Дорешивай задачи, будь как Саша!

В первом примере единственная смешная пара — это $(2, 5)$, так как $2 \oplus 3 = 4 \oplus 5 = 1$.

Во втором примере смешные пары $(2, 3)$, $(1, 4)$, и $(3, 6)$.

В третьем примере нет смешных пар.

A. Tokitsukaze и странное неравенство

ограничение по времени на тест: 1.5 секунд

ограничение по памяти на тест: 256 мегабайт

У Tokitsukaze есть перестановка p длины n . Напомним, что перестановкой p длины n называется последовательность p_1, p_2, \dots, p_n , состоящая из n различных целых чисел, каждое из которых от 1 до n ($1 \leq p_i \leq n$).

Она хочет знать, сколько различных наборов индексов $[a, b, c, d]$ ($1 \leq a < b < c < d \leq n$) в этой перестановке удовлетворяют следующим двум неравенствам:

$$p_a < p_c \text{ и } p_b > p_d.$$

Обратите внимание, что два набора $[a_1, b_1, c_1, d_1]$ и $[a_2, b_2, c_2, d_2]$ считаются различными, если $a_1 \neq a_2$, или $b_1 \neq b_2$, или $c_1 \neq c_2$, или $d_1 \neq d_2$.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Каждый набор входных данных состоит из двух строк.

В первой строке записано одно целое число n ($4 \leq n \leq 5000$) — длина перестановки p .

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — перестановка p .

Гарантируется, что сумма n по всем тестам не превышает 5000.

Выходные данные

Для каждого набора входных данных выведите одно целое число — количество различных наборов индексов $[a, b, c, d]$.

Пример

входные данные	Скопировать
3 6 5 3 6 1 4 2 4 1 2 3 4 10 5 1 6 2 8 3 4 10 9 7	
выходные данные	Скопировать
3 0 28	

Примечание

В первом наборе входных данных имеется 3 различных набора $[a, b, c, d]$.

$p_1 = 5, p_2 = 3, p_3 = 6, p_4 = 1$, где $p_1 < p_3$ и $p_2 > p_4$ удовлетворяет неравенству, поэтому один из $[a, b, c, d]$ наборов — это $[1, 2, 3, 4]$.

Точно так же два других набора — это $[1, 2, 3, 6]$ и $[2, 3, 5, 6]$.



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

D. Почти тождественные перестановки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Перестановка p размера n — такой массив, что каждое целое число от 1 до n встречается ровно один раз в этом массиве.

Будем называть *почти тождественными* такие перестановки, что для каждой из них существует как минимум $n - k$ индексов i ($1 \leq i \leq n$), таких, что $p_i = i$.

Вам необходимо подсчитать количество *почти тождественных* перестановок для заданных n и k .

Входные данные

В первой строке записаны два числа — n и k ($4 \leq n \leq 1000$, $1 \leq k \leq 4$).

Выходные данные

Выведите число *почти тождественных* перестановок для заданных n и k .

Примеры

входные данные	Скопировать
4 1	
выходные данные	Скопировать
1	
входные данные	Скопировать
4 2	
выходные данные	Скопировать
7	
входные данные	Скопировать
5 3	
выходные данные	Скопировать
31	
входные данные	Скопировать
5 4	
выходные данные	Скопировать
76	

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:50:29 UTC+5 (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ITMO

D. Радиовышки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

На координатной прямой расположено $n + 2$ города, с номерами от 0 до $n + 1$. i -й город расположен в точке i .

В каждом из городов $1, 2, \dots, n$ с вероятностью $\frac{1}{2}$ строится радиовышка (эти события независимы). После этого вы хотите установить мощность сигнала каждой вышки равной целому числу от 1 до n (мощность сигнала не обязательно одинакова для всех вышек, но и не обязательно отличается). Сигнал с вышки, расположенной в городе i с мощностью сигнала p , достигает каждого города c , что $|c - i| < p$.

После постройки вышек вы хотите выбрать мощность сигнала таким образом, чтобы:

- города 0 и $n + 1$ не получали сигнал от радиовышек;
- города $1, 2, \dots, n$ получали сигнал ровно от одной радиовышки каждый.

Например, если $n = 5$, и вышки построены в городах 2, 4 и 5, вы можете установить мощность сигнала вышки в городе 2 равной 2, а мощность сигнала вышек в городах 4 и 5 равной 1. Таким образом, города 0 и $n + 1$ не получат сигнал ни от одной вышки, города 1, 2 и 3 получат сигнал от вышки в городе 2, город 4 получит сигнал от вышки в городе 4, а город 5 получит сигнал от вышки в городе 5.

Посчитайте вероятность того, что после постройки вышек у вас будет возможность установить мощность сигнала каждой из них, чтобы удовлетворить всем ограничениям.

Входные данные

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

Выходные данные

Выведите одно целое число — вероятность того, что найдется способ установить мощность сигнала так, чтобы все ограничения были выполнены, взятая по модулю 998244353.

Вероятность можно выразить в виде несократимой дроби $\frac{x}{y}$. Вы должны вывести значение $x \cdot y^{-1} \pmod{998244353}$, где y^{-1} — целое число, такое, что $y \cdot y^{-1} \pmod{998244353} = 1$.

Примеры

входные данные	<input type="button" value="Скопировать"/>
2	
выходные данные	<input type="button" value="Скопировать"/>
748683265	

входные данные	<input type="button" value="Скопировать"/>
3	
выходные данные	<input type="button" value="Скопировать"/>
748683265	

входные данные	<input type="button" value="Скопировать"/>
5	
выходные данные	<input type="button" value="Скопировать"/>
842268673	

входные данные	<input type="button" value="Скопировать"/>
200000	
выходные данные	<input type="button" value="Скопировать"/>
202370013	

Примечание

Настоящий ответ для первого примера — $\frac{1}{4}$:

- с вероятностью $\frac{1}{4}$ вышки построены в обоих городах 1 и 2, поэтому мы можем установить их мощность сигнала равной 1.

Настоящий ответ для второго примера — $\frac{1}{4}$:

- с вероятностью $\frac{1}{8}$ вышки построены в городах 1, 2 и 3, поэтому мы можем установить их мощность сигнала равной 1;
- с вероятностью $\frac{1}{8}$ построена только одна башня в городе 2, и мы можем установить ее мощность сигнала равной 2.

Настоящий ответ для третьего примера — $\frac{5}{32}$. Обратите внимание, что даже если предыдущие пояснения использовали равные мощности сигнала для всех башен, это не обязательно так. Например, если $n = 5$ и башни построены в городах 2, 4 и 5, вы можете установить мощность сигнала башни в городе 2 равной 2, а мощность сигнала башен в городах 4 и 5 равной 1.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.11.2024 21:50:32^{UTC+5} (j1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке




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

C. Звёздное небо

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На небе задана декартова система координат. Там видно n звёзд, i -я имеет координаты (x_i, y_i) , максимальную яркость c , общую для всех звёзд, и начальную яркость s_i ($0 \leq s_i \leq c$).

С течением времени звёзды мерцают. В момент времени 0 i -я звезда имеет яркость, равную начальной s_i . Пусть в момент времени t какая-то звезда имеет яркость x . Тогда в $(t + 1)$ -й момент времени эта звезда будет иметь яркость $x + 1$, если $x + 1 \leq c$, а иначе у неё будет яркость 0.

Вы хотите q раз посмотретьеть на небо. В i -й раз вы посмотрите в момент времени t_i и увидите прямоугольник, стороны которого параллельны осям координат, левый нижний угол имеет координаты (x_{1i}, y_{1i}) , а правый верхний — (x_{2i}, y_{2i}) . Для каждого просмотра вы хотите узнать суммарную яркость звёзд, лежащих в просмотренном прямоугольнике.

Звезда лежит в прямоугольнике, если она лежит на его границе, либо лежит строго внутри него.

Входные данные

Первая строка содержит три целых числа n, q, c ($1 \leq n, q \leq 10^5$, $1 \leq c \leq 10$) — количество звёзд, количество просмотров неба и максимальная яркость звёзд.

Следующие n строк содержат описание звёзд. i -я из этих строк содержит три целых числа x_i, y_i, s_i ($1 \leq x_i, y_i \leq 100$, $0 \leq s_i \leq c \leq 10$) — координаты i -й звезды и её начальная яркость.

Следующие q строк содержат описание участков неба. i -я из этих строк содержит пять целых чисел $t_i, x_{1i}, y_{1i}, x_{2i}, y_{2i}$ ($0 \leq t_i \leq 10^9$, $1 \leq x_{1i} < x_{2i} \leq 100$, $1 \leq y_{1i} < y_{2i} \leq 100$) — момент времени i -го просмотра и координаты просмотренного прямоугольника.

Выходные данные

Для каждого просмотра выведите суммарную яркость просмотренных звёзд.

Примеры

входные данные	Скопировать
2 3 3 1 1 1 3 2 0 2 1 1 2 2 0 2 1 4 5 5 1 1 5 5	
выходные данные	Скопировать
3 0 3	
входные данные	Скопировать
3 4 5 1 1 2 2 3 0 3 3 1 0 1 1 100 100 1 2 2 4 4 2 2 1 4 7 1 50 50 51 51	
выходные данные	Скопировать
3 3 5 0	

Примечание

Рассмотрим первый пример.

При первом просмотре видно только первую звезду. В момент времени 2 её яркость равна 3, поэтому ответ на этот запрос — 3.


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

A. Нестандартное мышление

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Кевин только что получил свои неутешительные результаты по Олимпиаде по Идентификации Коров в США (USAICO) в виде бинарной строки длины n . Каждый символ строки Кевина представляет собой результат Кевина на одном из n вопросов олимпиады — '1' если корова была определена правильно и '0' в противном случае.

Тем не менее не всё потеряно. Кевин — большой любитель нестандартного мышления, и он считает, что его результат должен быть не суммой его очков, а максимальной длиной чередующейся подпоследовательности его строки. Здесь мы определяем чередующуюся подпоследовательность строки как подпоследовательность (необязательно из соседних элементов), в которой никакие два последовательных элемента не равны. Например, {0, 1, 0, 1}, {1, 0, 1}, и {1, 0, 1, 0} — чередующиеся подпоследовательности, в то время как {1, 0, 0} и {0, 1, 0, 1, 1} таковыми не являются.

Кевин, будучи ещё тем хитрецом, готов взломать базу данных USAICO, чтобы улучшить свой результат. Для того чтобы провернуть дело незаметно, он решил, что инвертирует ровно одну подстроку, то есть возьмёт несколько **последовательных** элементов строки своего результата и изменит все '0' в этой подстроке на '1' и наоборот. Кевин хочет знать максимально возможную длину самой длинной чередующейся подпоследовательности, которую может иметь строка с его результатами после одной операции инвертирования.

Входные данные

Первая строка входных данных содержит единственное число n ($1 \leq n \leq 100\,000$) — количество вопросов на олимпиаде.

Следующая строка содержит бинарную строку длины n , задающую результаты Кевина на USAICO.

Выходные данные

Выведите одно целое число — длину самой длинной возможной чередующейся подпоследовательности, которую Кевин может создать в своей строке, **инвертировать** ровно одну подстроку.

Примеры

входные данные	Скопировать
8 10000011	
выходные данные	Скопировать
5	
входные данные	Скопировать
2 01	
выходные данные	Скопировать
2	

Примечание

В первом примере Кевин может **инвертировать** выделенную жирным подстроку '10000011' и превратить свою строку в '10011011', которая имеет переменную подпоследовательность длины 5: '10011011'.

Во втором примере Кевин может **инвертировать** всю строку, и это не изменит результат.



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

C. Обычная условно-бесплатная игра

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вы играете в игру, в которой вашему персонажу нужно преодолевать различные препятствия. Например, сейчас вам нужно спуститься с утеса. Высота утеса равна h , и на каждой целой высоте x от 1 до h есть выдвижная платформа.

Каждая платформа либо спрятана в утесе, либо выдвинута. Изначально выдвинуто n платформ на высотах p_1, p_2, \dots, p_n . Платформа на высоте h выдвинута (и на ней изначально стоит персонаж).

Если ваш персонаж стоит на выдвинутой платформе на высоте x , то он может нажать на специальный рычаг, который меняет положение **двух платформ: на высотах x и $x - 1$** . Другими словами, платформа, на которой вы стоите, спрячется в утес, а платформа ниже поменяет свое состояние: если она была выдвинута, она спрятается, а если она была спрятана — выдвинется. Во втором случае вы безопасно приземлитесь на нее. **Заметьте, что нажать на рычаг — это единственный способ спуститься с платформы.**

Ваш персонаж довольно хрупок, так что он может безопасно падать с высоты не более 2. Другими словами, падение с платформы x на платформу $x - 2$ безопасно, но падение с платформы x на платформу $x - 3$ (или ниже) смертельно.

Иногда невозможно безопасно спуститься с утеса, но вы всегда можете купить (естественно за реальные деньги) несколько магических кристаллов. Каждый магический кристалл может быть использован для изменения состояния любой платформы (за исключением платформ на высоте h). После использования кристалл исчезает.

Какое минимальное количество кристаллов вам нужно купить для того, чтобы безопасно спуститься на землю, которая находится на высоте 0?

Входные данные

Первая строка содержит число q ($1 \leq q \leq 100$) — количество запросов. Каждый запрос состоит из двух строк.

Первая строка каждого запроса содержит два числа h и n ($1 \leq h \leq 10^9$, $1 \leq n \leq \min(h, 2 \cdot 10^5)$) — высота утеса и количество выдвинутых платформ.

Вторая строка каждого запроса содержит n чисел p_1, p_2, \dots, p_n ($h = p_1 > p_2 > \dots > p_n \geq 1$) — выдвинутые платформы в порядке убывания их высоты.

Сумма всех n по всем запросам не превосходит $2 \cdot 10^5$.

Выходные данные

На каждый запрос выведите одно число — минимально возможное количество кристаллов, которое вам нужно купить для безопасного спуска на землю (которая находится на высоте 0).

Пример

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


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

C. Раскройте скобки

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

У RSJ есть последовательность a из n целых чисел a_1, a_2, \dots, a_n и целое число s . Для каждого a_2, a_3, \dots, a_{n-1} он выбрал пару **неотрицательных целых чисел** x_i и y_i такую, что $x_i + y_i = a_i$ и $(x_i - s) \cdot (y_i - s) \geq 0$.

Теперь он заинтересовался значением

$$F = a_1 \cdot x_2 + y_2 \cdot x_3 + y_3 \cdot x_4 + \dots + y_{n-2} \cdot x_{n-1} + y_{n-1} \cdot a_n.$$

Помогите ему найти минимально возможное значение F при оптимальном выборе x_i и y_i . Можно показать, что всегда существует хотя бы один корректный способ выбрать эти числа.

Входные данные

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

Первая строка каждого набора входных данных содержит два целых числа n, s ($3 \leq n \leq 2 \cdot 10^5$; $0 \leq s \leq 2 \cdot 10^5$).

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 2 \cdot 10^5$).

Гарантируется, что сумма n не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите минимальное значение F .

Пример

входные данные	Скопировать
<pre>10 5 0 2 0 1 3 4 5 1 5 3 4 3 5 7 2 7 6 5 4 3 2 1 5 1 1 2 3 4 5 5 2 1 2 3 4 5 4 0 0 1 1 1 5 5 4 3 5 6 4 4 1 0 2 1 0 3 99999 200000 200000 200000 6 8139 7976 129785 12984 78561 173685 15480</pre>	
выходные данные	Скопировать
<pre>0 18 32 11 14 0 16 0 40000000000 2700826806</pre>	

Примечание

В первом наборе входных данных $2 \cdot 0 + 0 \cdot 1 + 0 \cdot 3 + 0 \cdot 4 = 0$.

Во втором наборе входных данных $5 \cdot 1 + 2 \cdot 2 + 2 \cdot 2 + 1 \cdot 5 = 18$.


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

C. Пылкий поток любви

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

День рождения Надеко приближается! Так как она украсила комнату для вечеринки, длинная гирлянда из бумажных гвоздик была повешена на видную часть стены. Койому, брату Надеко, она понравится!

Надеке все еще не нравится гирлянда, поэтому она решила ее снова улучшить. Гирлянда состоит из n частей, пронумерованных слева направо от 1 до n , и i -я из этих частей имеет цвет s_i , обозначенный строчной латинской буквой. Надеко перекрасит **не более** m частей в любой цвет (по-прежнему обозначенный строчной латинской буквой). После этого, она найдет все подотрезки, состоящие только из частей цвета c — любимого цвета Койоми — и будет называть длину самого большого такого подотрезка койомностью гирлянды.

Пусть, например, гирлянда имеет цвета «коомто», а любимый цвет Койоми — «о». Среди всех подотрезков, содержащих только цвет «о», самым длинным является «ooo», его длина равна 3. Поэтому койомность этой гирлянды равна 3.

Проблема в том, что Надеко не уверена в любимом цвете Койоми, поэтому она постоянно меняет свои планы на предстоящую работу. У нее есть q планов, каждый определяется парой из числа m_i и строчной буквы c_i , значения которых были объяснены раньше. Вам предстоит найти максимальную койомность, достижимую после изменения гирлянды в соответствии с каждым из планов.

Входные данные

Первая строка содержит одно положительное целое число n ($1 \leq n \leq 1\,500$) — длину гирлянды.

Вторая строка содержит n строчных латинских букв $s_1 s_2 \dots s_n$ как строку — начальные цвета частей гирлянды.

Третья строка содержит целое положительное число q ($1 \leq q \leq 200\,000$) — количество планов у Надеко.

Каждая из следующих q строк описывает один план: строка i из них содержит целое число m_i ($1 \leq m_i \leq n$) — максимальное число частей, которое можно перекрасить, и затем строчную латинскую букву c_i — возможный любимый цвет Койоми.

Выходные данные

Выведите q строк: для каждого плана выведите одно целое число на отдельной строке — максимально возможную койомность, достижимую после изменения гирлянды в соответствии с этим планом.

Примеры

входные данные	Скопировать
<pre>6 коомти 3 1 o 4 o 4 m</pre>	
выходные данные	Скопировать
<pre>3 6 5</pre>	

входные данные	Скопировать
<pre>15 yamatonadeshiko 10 1 a 2 a 3 a 4 a 5 a 1 b 2 b 3 b 4 b 5 b</pre>	
выходные данные	Скопировать

```
3  
4  
5  
7  
8  
1  
2  
3  
4  
5
```

входные данные**Скопировать**

```
10  
aaaaaaaaaa  
2  
10 b  
10 z
```

выходные данные**Скопировать**

```
10  
10
```

Примечание

В первом примере есть три плана:

- В первом плане можно перекрасить не более чем 1 часть. Перекрасив часть цвета «у» в цвет «о», получим «коооми», кожомность которой равна 3 — максимально возможной;
- Во втором плане можно перекрасить не более чем 4 части, получив гирлянду «oooooo», кожомность которой равна 6;
- В третьем случае можно перекрасить не более 4 частей, два способа «mmmmmi» и «kmmmm» являются оптимальными с кожомностью, равной 5.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.11.2024 21:50:49 UTC+5 (j1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО


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

C. Цветная полоска

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Цветная полоска представляет собой n расположенных в ряд клеточек, каждая из которых покрашена в один из k цветов. Требуется перекрасить наименьшее количество клеточек так, чтобы никакие две соседние клетки не были одинакового цвета. Перекрашивать клетки можно в любые цвета от 1 до k .

Входные данные

В первой строке входных данных записано два целых числа n и k ($1 \leq n \leq 5 \cdot 10^5$; $2 \leq k \leq 26$). Вторая строка состоит из n прописных букв латинского алфавита. Буква «A» обозначает первый цвет, «B» — второй и так далее. В строке используются только первые k букв латинского алфавита. Каждая буква обозначает цвет соответствующей клетки полоски.

Выходные данные

Выведите единственное число — искомое минимальное количество перекрашиваний. Во вторую строку выведите любой из возможных вариантов полоски после перекрашиваний.

Примеры

входные данные	Скопировать
6 3 ABBACC	
выходные данные	Скопировать
2 ABCASA	
входные данные	Скопировать
3 2 BBB	
выходные данные	Скопировать
1 BAB	

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:50:53 UTC+5 (j1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО


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

В. Длинный путь

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Однажды мальчик Вася оказался в лабиринте, состоящем из $(n + 1)$ комнат, пронумерованных от 1 до $(n + 1)$. Первоначально Вася находится в первой комнате, а чтобы выйти из лабиринта, нужно попасть в $(n + 1)$ -ю.

Лабиринт устроен следующим образом. В каждой комнате лабиринта есть два односторонних портала. Рассмотрим комнату с номером i ($1 \leq i \leq n$), с помощью первого портала можно перейти из нее в комнату с номером $(i + 1)$, с помощью второго можно перейти из нее в комнату с номером p_i , где $1 \leq p_i \leq i$.

Чтобы не заблудиться Вася решил действовать следующим образом.

- Каждый раз когда Вася приходит в какую-то комнату он рисует на ее потолке крестик. Изначально, Вася рисует крестик на потолке комнаты 1.
- Пусть Вася находится в комнате i и уже нарисовал крестик на ее потолке. Тогда, если сейчас на потолке нарисовано нечетное количество крестиков, Вася будет пользоваться вторым порталом (он ведет в комнату p_i), иначе Вася будет пользоваться первым порталом.

Помогите Васе определить, сколько раз ему придется воспользоваться порталами, чтобы попасть в комнату $(n + 1)$ в конечном итоге.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^3$) — число комнат. Во второй строке записаны n целых чисел p_i ($1 \leq p_i \leq i$) — номера комнат, в которые можно попасть из i -й комнаты, если пользоваться вторым порталом.

Выходные данные

Выведите одно число — количество перемещений, за которое мальчик выберется из лабиринта. Поскольку это число может быть достаточно большим, выведите его по модулю 1000000007 ($10^9 + 7$).

Примеры

входные данные	Скопировать
2 1 2	
выходные данные	Скопировать
4	
входные данные	Скопировать
4 1 1 2 3	
выходные данные	Скопировать
20	
входные данные	Скопировать
5 1 1 1 1 1	
выходные данные	Скопировать
62	

C. Рыцари круглого стола

ограничение по времени на тест: 0.5 second

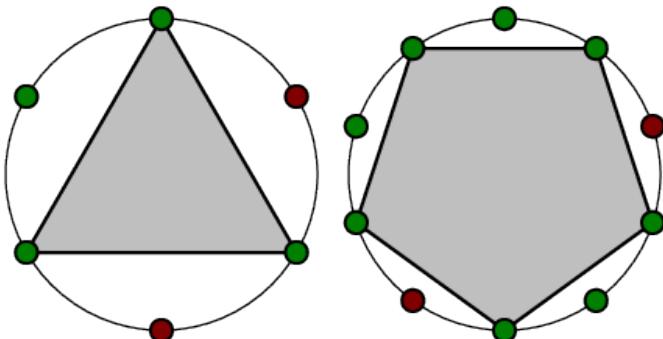
ограничение по памяти на тест: 256 megabytes

За круглым столом сидят n рыцарей на одинаковом расстоянии друг от друга. Каждый из них пребывает либо в хорошем настроении, либо в плохом.

Волшебник Мерлин предсказал королю Артуру, что грядущий месяц окажется особенно удачливым, если найдется такой правильный многоугольник, что во всех его вершинах находятся рыцари в хорошем настроении. Иначе грядущий месяц сулит неудачи.

Выпуклый многоугольник называется правильным, если все его стороны равны, а так же равные все его углы. В этой задаче мы считаем правильными только многоугольники хотя бы с 3 вершинами, то есть невырожденные.

На рисунке ниже приведены примеры таких многоугольников. Зеленые точки означают рыцарей в хорошем настроении, а красные - в плохом.



Королю Артуру известны настроения всех рыцарей. Помогите ему узнать — будет следующий месяц удачливым или нет.

Входные данные

В первой строке находится число n — количество рыцарей за круглым столом ($3 \leq n \leq 10^5$). Во второй строке через пробел перечислены настроения всех n рыцарей в порядке обхода стола. «1» означает, что рыцарь пребывает в хорошем настроении, «0» — что в плохом.

Выходные данные

Выведите «YES» без кавычек, если следующий месяц окажется удачливым. Иначе выведите «NO».

Примеры

входные данные	Скопировать
3 1 1 1	
выходные данные	Скопировать
YES	
входные данные	Скопировать
6 1 0 1 1 1 0	
выходные данные	Скопировать
YES	
входные данные	Скопировать
6 1 0 0 1 0 1	
выходные данные	Скопировать
NO	


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

C. Игра с перестановкой

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

После долгого дня Алиса и Боб решили сыграть в игру. Игровая доска состоит из n ячеек на прямой, пронумерованных от 1 до n , где каждая ячейка содержит число a_i в диапазоне от 1 до n . Кроме того, никакие две ячейки не содержат одинаковые числа.

Игровая фигурка находится на одной ячейке. Они ходят по очереди, двигая фигуруку. Алиса ходит первая. Игрок, который ходит, может передвинуть фигуруку с ячейки i на ячейку j , только если выполняются условия:

- число в новой ячейке j строго больше, чем в старой ячейке i (то есть $a_j > a_i$), и
- расстояние, на которое передвигается фигурука в этом ходу, должно делиться на число, расположеннное в старой ячейке (то есть $|i - j| \bmod a_i = 0$).

Проигрывает тот, кто не может сделать ход. Для каждой начальной позиции нужно определить, кто победит при оптимальной игре обоих. Можно доказать, что игра всегда когда-нибудь закончится, то есть всегда есть оптимальная стратегия для одного игрока.

Входные данные

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

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$). Гарантируется, что нет таких чисел $i \neq j$, что $a_i = a_j$.

Выходные данные

Выведите s — строку из n символов, где i -й символ представляет результат игры, которая начинается в ячейке i . Если победит Алиса, то s_i должен быть равен «A»; в другом случае, s_i должен быть равен «B».

Примеры

входные данные	Скопировать
8 3 6 5 4 2 7 1 8	
выходные данные	Скопировать
ВААААВАВ	
входные данные	Скопировать
15 3 11 2 5 10 9 7 13 15 8 4 12 6 1 14	
выходные данные	Скопировать
АВААААВВВААВААВ	

Примечание

В первом примере, если Боб поместит игровую фигурку на число (не на позицию):

- 1: Алиса может переместиться на любую ячейку. Она может выбрать число 7, из которого у Боба нет выхода.
- 2: Алиса может переместиться на числа 3 и 5. При переходе в 5, Боб может выиграть, если переместится в 8. Если она выберет 3, то она победит, так как Боб сможет переместиться только в 4, из которого Алиса сможет переместиться в 8.
- 3: Алиса может переместиться только в 4, после чего Боб побеждает, перемещаясь в 8.
- 4, 5 или 6: Алиса побеждает, перемещаясь в 8.
- 7, 8: Алиса не может никуда переместиться, поэтому проигрывает.


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

B. Ресторан

ограничение по времени на тест: 4 секунды

ограничение по памяти на тест: 256 мегабайт

Ресторан получил n заказов на проведение банкета. Каждый заказ характеризуется двумя величинами: моментом начала банкета l_i и моментом конца r_i ($l_i \leq r_i$).

Руководство ресторана может либо принять заказ, либо отвергнуть его. Какое наибольшее количество заказов может быть принято?

Никакие два принятых заказа не могут пересекаться, то есть не должно существовать момента времени, который принадлежит сразу двум принятым заказам. Если один из заказов начинается в момент, когда заканчивается другой, то они не могут быть приняты вместе.

Входные данные

В первой строке находится целое число n ($1 \leq n \leq 5 \cdot 10^5$) — количество заказов. В каждой из следующих n строк находится пара целых чисел l_i, r_i ($1 \leq l_i \leq r_i \leq 10^9$).

Выходные данные

Выведите наибольшее количество заказов, которые могут быть приняты.

Примеры

входные данные	Скопировать
2	
7 11	
4 7	
выходные данные	Скопировать
1	
входные данные	Скопировать
5	
1 2	
2 3	
3 4	
4 5	
5 6	
выходные данные	Скопировать
3	
входные данные	Скопировать
6	
4 8	
1 5	
4 7	
2 5	
1 3	
6 8	
выходные данные	Скопировать
2	


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

E. Любимая двоичная строка Качины

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Это интерактивная задача.

Качина бросает вам вызов угадать её любимую двоичную строку* s длины n . Она определяет $f(l, r)$ как количество подпоследовательностей† 01 в $s_l s_{l+1} \dots s_r$. Две подпоследовательности считаются различными, если они образованы путём удаления символов из разных позиций в исходной строке, даже если получившиеся подпоследовательности состоят из одних и тех же символов.

Чтобы определить s , вы можете задать ей несколько вопросов. В каждом вопросе вы можете выбрать два индекса l и r ($1 \leq l < r \leq n$) и спросить её о значении $f(l, r)$.

Определите и выведите s , задав Качине не более n вопросов. Однако может случиться так, что s невозможно определить. В этом случае вам нужно будет вывести **IMPOSSIBLE** вместо этого.

Формально, s считается **невозможно определить**, если после того, как вы задали n вопросов, всегда существует несколько возможных строк для s , независимо от того, какие вопросы задаются. **Обратите внимание, что если вы сообщите IMPOSSIBLE в то время, когда существует последовательность из не более чем n запросов, которая уникально определит двоичную строку, вы получите вердикт Неправильный ответ.**

* Двоичная строка содержит только символы 0 и 1.

† Последовательность a является подпоследовательностью последовательности b , если a может быть получена из b путём удаления нескольких (возможно, нуля или всех) элементов. Например, подпоследовательностями 1011101 являются 0 , 1 , 1111 , 0111 , но не 000 и не 11100 .

Входные данные

Первая строка ввода содержит одно целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 10^4$) — длина s .

Гарантируется, что сумма n по всем наборам входных данных не превышает 10^4 .

Протокол взаимодействия

Чтобы задать вопрос, выведите строку в следующем формате (не включайте кавычки)

- «? l r » ($1 \leq l < r \leq n$)

Жюри вернёт целое число $f(l, r)$.

Когда вы будете готовы напечатать ответ, выведите одну строку в следующем формате

- Если s невозможно определить, выведите «! **IMPOSSIBLE**»
- В противном случае выведите «! s »

После этого продолжайте обрабатывать следующий набор входных данных или завершите программу, если это был последний набор входных данных. Печать ответа не считается запросом.

Интерактор **не** адаптивен, что означает, что ответ известен до того, как участник задаст запросы и не зависит от заданных участником запросов.

Если ваша программа делает более n запросов для одного набора входных данных, ваша программа должна немедленно завершиться, чтобы получить вердикт **Неправильный ответ**. В противном случае вы можете получить произвольный вердикт, потому что ваше решение продолжит читать из закрытого потока.

После печати запроса не забудьте вывести конец строки и сбросить вывод. В противном случае вы можете получить вердикт **Превышен лимит бездействия**. Для этого используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Взломы

Чтобы сделать Взлом, используйте следующий формат.

Первая строка должна содержать одно целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных.

Первая строка каждого набора входных данных должна содержать целое число n ($2 \leq n \leq 10^4$) — длина s .

Следующая строка должна содержать s , двоичную строку длины n .

Сумма n по всем наборам входных данных не должна превышать 10^4 .

Пример

входные данные	Скопировать
2 5 4 0 1 2 2 0	
выходные данные	Скопировать
? 1 5 ? 2 4 ? 4 5 ? 3 5 ! 01001 ? 1 2 ! IMPOSSIBLE	

Примечание

В первом наборе входных данных:

В первом запросе вы спрашиваете Качину о значении $f(1, 5)$, и она отвечает 4 в потоке ввода.

Во втором запросе вы спрашиваете Качину о значении $f(2, 4)$. Поскольку в строке 100 нет подпоследовательностей 01, она отвечает 0 в потоке ввода.

После того, как вы задали 4 вопроса, вы сообщаете 01001 как s , и это правильно.

Во втором наборе входных данных:

В первом запросе вы спрашиваете Качину о значении $f(1, 2)$, и она отвечает 0 в потоке ввода. Обратите внимание, что это единственный различный вопрос, который вы можете задать.

Однако обратите внимание, что строки 00 и 11 обе имеют ответ 0, и невозможно различить между ними. Поэтому мы сообщаем IMPOSSIBLE.

Обратите внимание, что этот пример служит только для демонстрации формата взаимодействия. Он не гарантирует, что предоставленные запросы являются оптимальными или однозначно определяют ответ. Тем не менее, можно показать, что существует последовательность из не более чем 5 запросов, которая однозначно определяет пример теста 1.



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

B. Единая карта

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В Байтсбурге вводится новая система оплаты общественного транспорта. Теперь будет единая карта для поездок на всем транспорте. При каждой поездке пассажир прикладывает карту к считывателю, и с него списывается сумма в соответствии с тарифом.

Тариф устроен следующим способом. Есть три вида билетов:

1. билет на одну поездку, стоит 20 байтландских рублей,
2. билет на 90 минут, стоит 50 байтландских рублей,
3. билет на сутки (1440 минут), стоит 120 байтландских рублей.

Уточним, что билет на x минут, активированный во время t , действует на поездки, стартовое время которых находится в диапазоне от t до $t + x - 1$, включительно. Считайте, что каждая поездка занимает ровно одну минуту.

Чтобы пассажиру было проще, тариф выбирается автоматически: при совершении каждой поездки система анализирует все прошедшие поездки и текущую и выбирает набор билетов на эти поездки с минимальной суммарной стоимостью. Пусть минимальная стоимость, за которую можно совершить поездки с первой по текущую, равна a , а сумма, списанная с пассажира за все предыдущие поездки, равна b . Тогда система списывает с пассажира разницу $a - b$.

Вам предстоит написать программу, которая по данным о времени совершенных пассажиром поездок рассчитает сумму, которую нужно с него снять после каждой из них.

Входные данные

В первой строке ввода содержится число n ($1 \leq n \leq 10^5$) — число поездок, которые сделал пассажир.

В каждой из следующих n строк содержится время начала поездки t_i ($0 \leq t_i \leq 10^9$), измеренное в минутах от момента запуска системы. Все t_i различны, заданы в порядке возрастания, т. е. $t_{i+1} > t_i$ для всех $1 \leq i < n$.

Выходные данные

Выведите n чисел. Для каждой поездки выведите сумму, которую нужно снять с карты пассажира.

Примеры

входные данные	Скопировать
3 10 20 30	

выходные данные	Скопировать
20 20 10	

входные данные	Скопировать
10 13 45 46 60 103 115 126 150 256 516	

выходные данные	Скопировать
20 20 10 0 20 0	

0
20
20
10

Примечание

В первом примере система работает следующим образом: первую и вторую поездку дешевле всего оплатить двумя билетами на одну поездку, поэтому с пассажира снимается по 20 рублей, после третьей поездки система понимает, что дешевле было бы купить билет на 90 минут. Этот билет стоит 50 рублей, а пассажир к этому времени уже заплатил 40 рублей, таким образом с него нужно дополнительно взять 10 рублей.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.11.2024 21:56:42 UTC+5 (k1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



| ИТМО


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

A. Разноцветные шарики

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарп играет с красными и синими шариками. Он выложил в ряд слева направо n шариков. Оказалось, что выложенные шарики образуют зеброид.

Непустая последовательность красных и синих шариков называется зеброидом, если цвета шариков в этой последовательности чередуются. Например, последовательности (красный; синий; красный) и (синий) — зеброиды, а последовательность (красный; красный) — нет.

Теперь Поликарпа заинтересовало, сколькими способами можно выбрать из этой последовательности подпоследовательность, которая также будет зеброидом? Помогите ему решить эту задачку, найдите остаток от деления количества способов на 1000000007 ($10^9 + 7$).

Входные данные

В первой строке записано единственное целое число n ($1 \leq n \leq 10^6$) — количество шариков в последовательности Поликарпа.

Выходные данные

Выведите единственное целое число — ответ на задачу по модулю 1000000007 ($10^9 + 7$).

Примеры

входные данные	<input type="button" value="Скопировать"/>
3	
выходные данные	<input type="button" value="Скопировать"/>
6	
входные данные	<input type="button" value="Скопировать"/>
4	
выходные данные	<input type="button" value="Скопировать"/>
11	

Примечание

Рассмотрим первый тестовый пример. Пусть изначально у Поликарпа была последовательность (красный; синий; красный), тогда выбрать зеброид можно шестью способами:

- взять первый шарик;
- взять второй шарик;
- взять третий шарик;
- взять первый и второй шарики;
- взять второй и третий шарики;
- взять первый, второй и третий шарики.

Можно доказать, что если в качестве изначальной последовательности Поликарпа взять (синий; красный; синий), то количество способов не изменится.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.11.2024 21:56:44 UTC+5 (k1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО


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

D. Бег с препятствиями

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Спортсмен стартует из точки $x_{start} = 0$ и бежит в точку с координатой $x_{finish} = m$ (на прямой). Также спортсмен умеет прыгать — для прыжка он сначала должен взять разбег длины не менее s метров (при этом на протяжении этих s метров на его пути не должно быть препятствий), а после этого он может прыгнуть на длину не более d метров. Бегать и прыгать разрешено только в направлении слева направо. **Начинать и заканчивать** прыжок можно только в точках с целочисленными координатами, в которых **нет препятствий**. Чтобы преодолеть какое-то препятствие, нужно приземлиться в точке, которая находится строго правее этого препятствия.

На пути спортсмена есть n препятствий в точках с координатами x_1, x_2, \dots, x_n . Пробегать через препятствия нельзя, их можно только перепрыгивать. Перед вами стоит задача определить, сможет ли спортсмен добраться до финишной точки.

Входные данные

В первой строке входных данных следуют четыре целых числа n, m, s и d ($1 \leq n \leq 200\,000$, $2 \leq m \leq 10^9$, $1 \leq s, d \leq 10^9$) — количество препятствий на пути бегуна, координата финишной точки, длина разбега перед прыжком и максимальная длина прыжка соответственно.

Во второй строке следует последовательность из n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq m - 1$) — координаты препятствий. Гарантируется, что в стартовой и финишной точке нет препятствий, а также в одной точке не может быть более одного препятствия. Координаты препятствий заданы в произвольном порядке.

Выходные данные

Если бегун не сможет добраться до финишной точки, выведите в первую строку выходных данных «IMPOSSIBLE» (без кавычек).

Если же спортсмен может добраться от старта до финиша, то выведите любой способ сделать это в следующем формате:

- выведите строку вида «RUN X» (где «X» должно быть целым положительным числом), если очередные «X» метров спортсмен должен бежать;
- выведите строку вида «JUMP Y» (где «Y» должно быть целым положительным числом), если спортсмен начинает прыжок и очередные «Y» метров должен находиться в полете.

Все команды «RUN» и «JUMP» должны строго чередоваться, начиная с «RUN», а также они должны быть выведены в хронологическом порядке. Не разрешается перепрыгивать финишную точку, но разрешается в ней приземляться после прыжка. Спортсмен должен остановиться, как только достигнет финиша.

Примеры

входные данные	<button>Скопировать</button>
3 10 1 3 3 4 7	
выходные данные	<button>Скопировать</button>
RUN 2 JUMP 3 RUN 1 JUMP 2 RUN 2	
входные данные	<button>Скопировать</button>
2 9 2 3 6 4	
выходные данные	<button>Скопировать</button>
IMPOSSIBLE	