

Задача А. Серега и веселье

Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Серега любит веселье. Только веселятся все по-разному. Серега веселится, решая задачи на запросы. Как-то Федя придумал такую задачу.

Задан массив a , состоящий n целых положительных чисел, и запросы к нему. Запросы бывают двух типов:

1. Сделать единичный циклический сдвиг вправо на отрезке от l до r (обе границы включительно). Другими словами, переставить элементы массива следующим образом:

$$a[l], a[l+1], \dots, a[r-1], a[r] \rightarrow a[r], a[l], a[l+1], \dots, a[r-1].$$

2. Посчитать количество чисел, равных k , на отрезке от l до r (обе границы включительно).

Федя побежал обрадовать Серегу новой задачей. Серега очень быстро справился с ней. Посмотрим, как справитесь вы?

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — количество элементов массива. Во второй строке записаны n целых чисел $a[1], a[2], \dots, a[n]$ ($1 \leq a[i] \leq n$) — элементы массива.

В третьей строке записано целое число q ($1 \leq q \leq 10^5$) — количество запросов. В следующих q строках заданы запросы.

Так как отвечать на запросы вы должны по мере их поступления, запросы будут **закодированы**. Запросы первого типа во входных данных заданы в формате: $1 \ l'_i \ r'_i$. Запросы второго типа во входных данных заданы в формате: $2 \ l'_i \ r'_i \ k'_i$. Все заданные числа целые и удовлетворяют ограничениям: $1 \leq l'_i, r'_i, k'_i \leq n$.

Чтобы из закодированных входных данных получить данные для запроса, нужно выполнить преобразования:

$$l_i = ((l'_i + lastans - 1) \bmod n) + 1; \quad r_i = ((r'_i + lastans - 1) \bmod n) + 1; \quad k_i = ((k'_i + lastans - 1) \bmod n) + 1.$$

Здесь $lastans$ обозначает последний ответ на запрос 2-го типа (изначально $lastans$ равно 0). Если значение l_i получилось больше r_i , то их нужно поменять местами.

Формат выходных данных

Для каждого запроса 2-го типа выведите ответ в отдельной строке.

Примеры

stdin	stdout
7 6 6 2 7 4 2 5 7 1 3 6 2 2 4 2 2 2 4 7 2 2 2 5 1 2 6 1 1 4 2 1 7 3	2 1 0 0
8 8 4 2 2 7 7 8 8 8 1 8 8 2 8 1 7 1 8 1 1 7 3 2 8 8 3 1 1 4 1 2 7 1 4 5	2 0

Задача В. Красные и синие лампочки

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

Есть n лампочек с номерами $1, 2, \dots, n$. Надо ровно m из них сделать красными, а все остальные — синими, чтобы максимизировать следующую величину: по всем $i \in [1, n - 1]$: если лампочки i и $i + 1$ разного цвета, то добавить к ответу a_i .

Формат входных данных

Первая строка содержит два числа — n и m ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq n - 1$).

Вторая строка содержит $n - 1$ число a_1, a_2, \dots, a_{n-1} ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите максимальную сумму, которую можно получить, если ровно m лампочек красные.

Примеры

стандартный ввод	стандартный вывод
6 2 3 1 4 1 5	11
7 6 2 7 1 8 2 8	10
11 7 12345 678 90123 45678901 234567 89012 3456 78901 23456 7890	46207983

Задача С. Величайший мельчайший циклический сдвиг

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.75 секунд
Ограничение по памяти: 256 мегабайт

Пусть для строки S , $f(S)$ - лексикографически минимальный ее циклический сдвиг. Например для $S = babca$, $f(s) = ababc$.

Вам даны 3 целых числа X, Y, Z . Вы хотите построить строку T , которая содержит ровно X букв 'a', Y букв 'b' и Z букв 'c'. Если таких строк несколько, то вы хотите построить ту, у которой $f(T)$ лексикографически максимально.

Найдите лексикографически максимальное значение, которое может принимать $f(T)$.

Формат входных данных

Единственная строка входных данных содержит 3 целых неотрицательных числа X, Y, Z . ($1 \leq X + Y + Z \leq 50$)

Формат выходных данных

Выведите одну строку - ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 1 1	acb
2 2 0	abab

Задача D. Минимизируй количество инверсий

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

Дана перестановка p длины n .

Можно выбрать любую её подпоследовательность, удалить её из перестановки и приписать в начало перестановки в том же порядке.

Для каждого k от 0 до n найдите минимальное количество инверсий в перестановке, которое можно получить, выбрав подпоследовательность длины ровно k .

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 50\,000$) — количество наборов входных данных.

В первой строке каждого набора вводится число n ($1 \leq n \leq 5 \cdot 10^5$) — длина перестановки.

Во второй строке каждого набора вводится перестановка p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$).

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

Формат выходных данных

Для каждого набора выведите $n + 1$ число, где i -е число — ответ на задачу для длины подпоследовательности, равной $i - 1$.

Пример

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

Замечание

Во втором наборе:

- Для длины 0: $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$: 4 инверсии.
- Для длины 1: $[4, 2, 1, 3] \rightarrow [1, 4, 2, 3]$: 2 инверсии.
- Для длины 2: $[4, 2, 1, 3] \rightarrow [2, 1, 4, 3]$ или $[4, 2, 1, 3] \rightarrow [1, 3, 4, 2]$: 2 инверсии.
- Для длины 3: $[4, 2, 1, 3] \rightarrow [2, 1, 3, 4]$: 1 инверсия.
- Для длины 4: $[4, 2, 1, 3] \rightarrow [4, 2, 1, 3]$: 4 инверсии.

Задача Е. Дамба

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

Есть изначально пустая дамба. Дамба может вместить не больше L литров воды.

Каждое утро в дамбу будет добавляться v_i литров воды температурой t_i градусов. Перед тем, как вода нальётся, можно вылить любое количество воды, которая сейчас находится в дамбе.

Для каждого дня требуется посчитать максимальную температуру воды в дамбе, если на момент этого дня в ней должно быть ровно L литров воды, а так же в любой момент времени дамба не должна переполниться.

Обратите внимание, что ответы не зависят друг от друга, то есть для того, чтобы максимизировать температуру в 3-й день не требуется, чтобы в 1-й первый или во 2-й день дамба содержала ровно L литров воды, но требуется, чтобы в 3-й день в ней было ровно L литров.

Если в дамбе находилось v_1 литров воды с температурой t_1 и в неё добавилось v_2 литров воды температурой t_2 , то итоговая температура будет равна $\frac{t_1 \cdot v_1 + t_2 \cdot v_2}{v_1 + v_2}$.

Формат входных данных

Первая строка входных данных содержит два числа n и L ($1 \leq n \leq 5 \cdot 10^5, 1 \leq L \leq 10^9$) — количество дней и вместимость дамбы.

Далее следуют n строк. i -я строка содержит два числа t_i и v_i ($0 \leq t_i \leq 10^9, 1 \leq v_i \leq L$) — температура и объём добавленной воды в i -й день.

Гарантируется, что $v_1 = L$.

Формат выходных данных

Выведите n чисел. i -е число должно быть равно максимальной температуре в i -й день. Погрешность ответа не должна превосходить 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 10 10 10 20 5 4 3	10 15.0000000000 13.2000000000
4 15 0 15 2 5 3 6 4 4	0 0.6666666667 1.8666666667 2.9333333333
4 15 1000000000 15 9 5 8 6 7 4	1000000000 666666669.6666666667 400000005.0000000000 293333338.8666666667

Задача F. Xor-подпоследовательность (сложная версия)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Это сложная версия задачи. Единственное различие состоит в том, что в этой версии $a_i \leq 10^9$.

Дан массив из n целых чисел $a_0, a_1, a_2, \dots, a_{n-1}$. Бряп захотел найти в данном массиве самую длинную **хорошую** подпоследовательность.

Массив $b = [b_0, b_1, \dots, b_{m-1}]$, где $0 \leq b_0 < b_1 < \dots < b_{m-1} < n$, будем называть подпоследовательностью длины m массива a .

Подпоследовательность $b = [b_0, b_1, \dots, b_{m-1}]$ длины m называется **хорошей**, если выполняется следующее условие:

- Для любого целого числа p ($0 \leq p < m - 1$) выполняется условие: $a_{b_p} \oplus b_{p+1} < a_{b_{p+1}} \oplus b_p$.

Так как Бряп не очень любознательная персона, он хочет знать лишь длину такой подпоследовательности. Помогите ему найти ответ на данную задачу.

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит единственное целое число n ($2 \leq n \leq 3 \cdot 10^5$) — длина массива.

Вторая строка каждого набора входных данных содержит n целых чисел a_0, a_1, \dots, a_{n-1} ($0 \leq a_i \leq 10^9$) — элементы массива.

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

Формат выходных данных

Для каждого набора входных данных единственное число — максимальную длину **хорошей** подпоследовательности.

Пример

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

Замечание

В первом наборе входных данных в качестве подпоследовательности мы можем выбрать оба элемента массива, так как $1 \oplus 1 < 2 \oplus 0$.

Во втором наборе входных данных мы можем взять элементы с индексами 1, 2 и 4 (в 0 нумерации). Для них выполняется: $2 \oplus 2 < 4 \oplus 1$ и $4 \oplus 4 < 1 \oplus 2$.