

Задача А. Жизни цветов важны

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

Пете на день рождения подарили очень необычный цветок, но Петя не всегда бывает дома и не всегда может его поливать. Вам даны n дней из жизни Пети, вы должны сказать, что станет с цветком в конце.

Цветок обладает следующими свойствами:

- Если цветок не полить два дня подряд, то он завянет.
- Если в i -й день цветок полили, то он вырастает на 1 сантиметр.
- Если цветок полили и в i -й, и в $(i-1)$ -й день ($i > 1$), то цветок растёт на 5 сантиметров вместо 1.
- Если в i -й день цветок не полили, то в этот день он не растёт.

На начало 1-го дня высота цветка равна 1 сантиметру. Чему равна высота цветка в конце n -го дня?

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее следуют наборы входных данных.

В первой строке каждого набора находится единственное целое число n ($1 \leq n \leq 100$).

Во второй строке набора находятся n целых чисел a_1, a_2, \dots, a_n ($a_i = 0$ или $a_i = 1$). Если $a_i = 1$, то Петя полил цветок в i -й день, иначе не полил.

Формат выходных данных

Для каждого набора входных данных требуется вывести единственное число k — размер цветка после n дней, или -1 , если цветок завял.

Пример

стандартный ввод	стандартный вывод
4	3
3	7
1 0 1	-1
3	1
0 1 1	
4	
1 0 0 1	
1	
0	

Задача В. Выворачивание массива

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

Вам дан массив a длины n .

Определим операцию **выворачивания** массива. Положим значение $x = a_n$. Далее массив a делится на две части: левую и правую. Левая часть содержит элементы массива a , которые не больше x ($\leq x$), а правая — элементы массива a , строго большие x ($> x$). Относительный порядок элементов в частях остается таким же, как до выворачивания, то есть разделение выполняется стабильно. Затем части склеиваются: слева левая, справа правая.

Например, если a равен $[2, 4, 1, 5, 3]$, то выворачивание происходит так: $[2, 4, 1, 5, 3] \rightarrow [2, 1, 3], [4, 5] \rightarrow [2, 1, 3, 4, 5]$.

Мы будем применять операцию выворачивания к массиву a много раз. Можно показать, что после нескольких операций массив a больше не будет меняться. Выведите наименьшее число k такое, что после k выворачиваний массив больше не будет меняться.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Далее следуют наборы входных данных.

В первой строке каждого набора находится единственное целое число n ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке набора находятся n целых 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 $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных требуется вывести единственное число k — наименьшее количество выворачиваний, после которых массив не будет меняться.

Пример

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

Замечание

Рассмотрим первый пример.

- Первое выворачивание: $a = [1, 4, 2, 5, 3]$, $x = 3$. $[2, 4, 1, 5, 3] \rightarrow [2, 1, 3], [4, 5] \rightarrow [2, 1, 3, 4, 5]$.
- Второе и последующие выворачивания: $a = [2, 1, 3, 4, 5]$, $x = 5$. $[2, 1, 3, 4, 5] \rightarrow [2, 1, 3, 4, 5], [] \rightarrow [2, 1, 3, 4, 5]$. Эти выворачивания не меняют массив, поэтому ответ 1.

Рассмотрим второй пример.

- Первое выворачивание: $a = [5, 3, 2, 4, 1]$, $x = 1$. $[5, 3, 2, 4, 1] \rightarrow [1], [5, 3, 2, 4] \rightarrow [1, 5, 3, 2, 4]$.
- Второе выворачивание: $a = [1, 5, 3, 2, 4]$, $x = 4$. $[1, 5, 3, 2, 4] \rightarrow [1, 3, 2, 4], [5] \rightarrow [1, 3, 2, 4, 5]$.
- Третье и последующие выворачивания: $a = [1, 3, 2, 4, 5]$, $x = 5$. $[1, 3, 2, 4, 5] \rightarrow [1, 3, 2, 4, 5], [] \rightarrow [1, 3, 2, 4, 5]$. Эти выворачивания не меняют массив, поэтому ответ 2.

Задача С. Минимизируйте расстояние

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

На числовой прямой расположены n складов. Склад номер i находится в точке x_i для всех $1 \leq i \leq n$.

У вас есть n коробок с товарами, которые нужно доставить по одной в каждый из n складов. Вы и все n коробок изначально находитесь в начале координат, точке 0. Одновременно вы можете переносить не более k коробок. Вы должны брать по несколько коробок из начала координат, доставлять их по складам, и возвращаться в начало координат для следующей партии коробок.

Вычислите минимальное расстояние, которое вам нужно пройти, чтобы доставить все коробки по складам. Вы не обязаны возвращаться в начало координат после того, как доставите все коробки.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10\,500$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и k ($1 \leq k \leq n \leq 2 \cdot 10^5$).

Вторая строка каждого набора входных данных содержит n целых чисел x_1, x_2, \dots, x_n ($-10^9 \leq x_i \leq 10^9$). Может оказаться, что некоторые склады находятся в одинаковых точках на прямой.

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

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимальное расстояние, которое вы должны пройти, чтобы доставить все коробки по складам.

Пример

стандартный ввод	стандартный вывод
4	25
5 1	41
1 2 3 4 5	7
9 3	3000000000
-5 -10 -15 6 5 8 3 7 4	
5 3	
2 2 3 3 3	
4 2	
1000000000 1000000000 1000000000 1000000000	

Замечание

В первом примере вы можете носить только по одной коробке. Поэтому один из путей, минимизирующий расстояние, такой: $0 \rightarrow 2 \rightarrow 0 \rightarrow 4 \rightarrow 0 \rightarrow 3 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 5$, где каждый 0 обозначает, что вы идете в начало координат и забираете одну коробку, а каждое положительное число означает доставку коробки в склад в соответствующей координате. Общее расстояние равно 25. Существуют и другие оптимальные решения.

Во втором примере одно из оптимальных решение такое: $0 \rightarrow 6 \rightarrow 8 \rightarrow 7 \rightarrow 0 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 0 \rightarrow (-5) \rightarrow (-10) \rightarrow (-15)$, общее расстояние равно 41. Можно показать, что 41 — минимальное возможное расстояние в этом примере.

Задача D. Ещё одна задача на сортировку

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

У Пети есть массив целых чисел a_1, a_2, \dots, a_n . Ему нравятся только отсортированные массивы. К сожалению, имеющийся массив может быть произвольным, поэтому Петя хочет его отсортировать.

Петя любит придумывать различные усложнения, поэтому он хочет отсортировать массив, используя только 3-циклы. Более формально, за одну операцию он может выбрать 3 попарно различные позиции i, j и k ($1 \leq i, j, k \leq n$) и применить цикл ($i \rightarrow j \rightarrow k \rightarrow i$) к массиву a . Это действие одновременно положит значение a_i на позицию j , значение a_j на позицию k , и значение a_k на позицию i , а остальные элементы массива останутся неизменными.

Например, если массив a равен $[10, 50, 20, 30, 40, 60]$, и Петя выбирает $i = 2, j = 1$ и $k = 5$, то массив изменится на $[50, 40, 20, 30, 10, 60]$.

Петя может применять произвольное количество 3-циклов (возможно, ноль). Определите, может ли Петя отсортировать массив a , то есть сделать его неубывающим.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 5 \cdot 10^5$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка набора входных данных содержит единственное целое число n ($1 \leq n \leq 5 \cdot 10^5$) — длину массива a .

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$).

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

Формат выходных данных

Для каждого набора входных данных выведите «YES» (без кавычек), если Петя может отсортировать массив a с помощью 3-циклов, и «NO» (без кавычек) в противном случае. Вы можете выводить каждую букву в любом регистре (верхнем или нижнем).

Пример

стандартный ввод	стандартный вывод
7	YES
1	YES
1	NO
2	YES
2 2	NO
2	YES
2 1	YES
3	
1 2 3	
3	
2 1 3	
3	
3 1 2	
4	
2 1 4 3	

Замечание

В 6-м наборе входных данных Петя может применить 3-цикл $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$ для сортировки массива.

В 7-м наборе входных данных Петя может сначала применить $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$, получив $a = [1, 4, 2, 3]$. Затем он может применить $2 \rightarrow 4 \rightarrow 3 \rightarrow 2$ и окончательно отсортировать массив.

Задача E. Запросы частот

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

У Пети есть подвешенное дерево, на вершинах которого написаны целые числа. Корень — вершина 1. Вам нужно ответить на некоторые вопросы про это дерево.

Дерево — это связный ациклический граф. Подвешенное дерево имеет специальную вершину — корень. Родителем вершины v называется следующая вершина на кратчайшем пути от v до корня.

Каждый запрос задан тройкой целых чисел v , l , и k . Чтобы на него ответить вы должны проделать следующие шаги:

- Сначала, выпишите последовательность чисел, записанных на вершинах простого пути от v до корня (включая вершину v и сам корень).
- Посчитайте, сколько раз каждой число встречается в этой последовательности. Выкиньте из неё числа, которые встречаются меньше, чем l раз.
- Преобразуйте последовательность, удалив из неё дубликаты, и упорядочив числа в ней возрастанию количеств вхождений в изначальную последовательность. В случае совпадения таких количеств, два числа располагаются в произвольном относительном порядке.
- Ответом на запрос является число, стоящее k -м в получившейся последовательности. Важно отметить, что так как порядок определён неоднозначно, то в качестве ответа примется любое число, которое могло стоять на этой позиции. Также может получиться, что в результирующей последовательности меньше k , в таком случае ответом считается -1 .

Например, если последовательность чисел на пути от v до корня равна $[2, 2, 1, 7, 1, 1, 4, 4, 4, 4]$, $l = 2$ и $k = 2$, то ответ равен 1.

Ответьте, пожалуйста, на все вопросы про дерево.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^6$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка набора входных данных содержит одно целое число n , q ($1 \leq n, q \leq 10^6$) — количество вершин в дереве и количество запросов.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$), где a_i — число записанное на i -й вершине.

Третья строка содержит $n - 1$ целых чисел p_2, p_3, \dots, p_n ($1 \leq p_i \leq n$), где p_i — родитель вершины i . Гарантируется, что значения p задают корректное дерево.

Каждая из последующих q строк содержит по три целых числа v, l, k ($1 \leq v, l, k \leq n$) — описание вопросов.

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

Формат выходных данных

Для каждого из запросов, выведите ответ на него. Если ответов несколько, то выведите любой.

Пример

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

Задача F. Неравные соседи

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

Вам дан массив a_1, a_2, \dots, a_n длины n , состоящий из положительных целых чисел. Ваша задача — посчитать количество массивов целых чисел b_1, b_2, \dots, b_n длины n таких, что:

- $1 \leq b_i \leq a_i$ для всех i ($1 \leq i \leq n$), и
- $b_i \neq b_{i+1}$ для всех i ($1 \leq i \leq n - 1$).

Количество таких массивов может быть очень большим, поэтому выведите это число по модулю 998 244 353.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина массива a .
Во второй строке задано n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходных данных

В единственной строке выведите ответ по модулю 998 244 353.

Примеры

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

Замечание

В первом примере подходят массивы $[1, 2, 1]$ и $[2, 1, 2]$.
Во втором примере допустимы массивы $[1, 2]$, $[1, 3]$, $[2, 1]$ и $[2, 3]$.

Задача G. Браконьеры

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Алиса и Боб — два браконьера, которые рубят лес.

Лес это набор (возможно пустой) деревьев. Дерево — это связный граф без циклов. Подвешенное дерево имеет специальную вершину — корень. Родителем вершины v называется следующая вершина на кратчайшем пути от v до корня. Дети вершины v называются вершины, для которых v является родителем. Вершина называется листом, если у неё нет детей.

В этой задаче мы определим *глубину* вершины как количество вершин на простом пути от этой вершины до корня. *Рангом* дерева назовем минимальную глубину его листа.

Изначально дан непустой лес подвешенных деревьев. Алиса и Боб играют в игру на этом лесе. Они ходят по очереди, Алиса ходит первой. В начале каждого хода игрок выбирает дерево из леса. Далее игрок выбирает положительное целое число — *глубину разреза*, которая **не превосходит ранга** выбранного дерева. Затем игрок удаляет из дерева все вершины, чьи глубины меньше либо равны глубине разреза. Все остальные вершины разбиваются на набор подвешенных деревьев, корнем каждого становится вершина, имевшая наименьшую глубину в дереве до разреза. Все эти деревья добавляются в лес и игра продолжается.

Игрок проигрывает, если на момент начала его хода лес пуст.

Определите, может ли Алиса победить, если оба игрока играют оптимально.

Формат входных данных

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 5 \cdot 10^5$) — количество наборов входных данных. Далее следуют наборы входных данных.

Первая строка набора входных данных содержит одно целое число n ($1 \leq n \leq 5 \cdot 10^5$) — суммарное количество вершин в деревьях изначального леса.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($0 \leq p_i \leq n$) — описание леса. Если $p_i = 0$, то i -я вершина дерева является корнем, иначе p_i является родителем вершины i . Гарантируется, что p задает корректный лес подвешенных деревьев.

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

Формат выходных данных

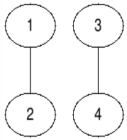
Для каждого набора входных данных выведите «YES» (без кавычек), если Алиса может победить, иначе выведите «NO» (без кавычек). Вы можете выводить каждую букву в любом регистре.

Пример

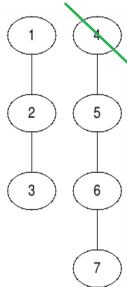
стандартный ввод	стандартный вывод
4	NO
4	YES
0 1 0 3	NO
7	YES
0 1 2 0 4 5 6	
4	
0 1 1 2	
7	
0 1 1 2 2 3 3	

Замечание

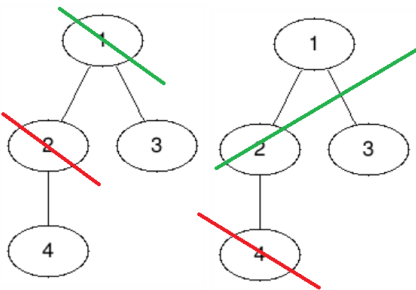
В первом наборе входных данных у Боба есть симметричная стратегия, поэтому Алиса выиграть не может.



Во втором наборе Алиса может выбрать второе дерево и глубину разреза 1, чтобы получить лес, где уже у нее есть симметричная стратегия.



В третьем наборе ранг единственного дерева равен 2, и оба возможных хода Алисы приводят к неудаче. В первом случае Боб может перейти к позиции с симметричной стратегией, а во втором сразу опустошить лес.



В четвертом наборе все листья единственного дерева имеют одинаковую глубину, поэтому Алиса может за один ход опустошить весь лес.

