

Задача А. RMQ

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

Реализуйте структуру данных, которая на данном массиве из N целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

Формат входных данных

В первой строке вводится натуральное число N ($1 \leq N \leq 10^5$) – количество элементов в массиве. В следующей строке содержатся N целых чисел, не превосходящих по модулю 10^9 – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число K ($0 \leq K \leq 10^5$) – количество запросов к структуре данных. Каждая из следующих K строк содержит два целых числа l и r ($1 \leq l \leq r \leq N$) – левую и правую границы отрезка в массиве для данного запроса.

Формат выходных данных

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от l до r и индекс одного из элементов массива, принадлежащий отрезку от l до r , на котором достигается этот максимум.

Пример

стандартный ввод	стандартный вывод
5	7 1
7 3 1 6 4	6 4
3	1 3
1 5	
2 4	
3 3	

Задача В. Нолики

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

Дедус любит давать своим ученикам сложные задачки. На этот раз он придумал такую задачу: Рейтинг всех его учеников записан в массив A . Запросы Дедуса таковы:

1. Изменить рейтинг i -го ученика на число x
2. Найти максимальную последовательность подряд идущих ноликов в массиве A на отрезке $[l, r]$.

Помогите бедным фиксикам избежать зверского наказания за нерешение задачи на этот раз.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 500\,000$) – количество учеников. Во второй строке записано N чисел – их рейтинги, числа по модулю не превосходящие 1000 (по количеству задач, которые ученик решил или не решил за время обучения). В третьей строке записано число M ($1 \leq M \leq 50\,000$) – количество запросов. Каждая из следующих M строк содержит описание запросов:

«UPDATE i x » – обновить i -ый элемент массива значением x ($1 \leq i \leq N$, $|x| \leq 1000$)

«QUERY l r » – найти длину максимальной последовательности из нулей на отрезке с l по r . ($1 \leq l \leq r \leq N$)

Формат выходных данных

В выходной файл выведите ответы на запросы «QUERY» в том же порядке, что и во входном файле

Пример

стандартный ввод	стандартный вывод
5	2
328 0 0 0 0	1
5	1
QUERY 1 3	
UPDATE 2 832	
QUERY 3 3	
QUERY 2 3	
UPDATE 2 0	

Задача С. Число возрастающих подпоследовательностей

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

Задана последовательность из n чисел $a_1, a_2, \dots, a_n$. Подпоследовательностью длины k этой последовательности называется набор индексов $i_1, i_2, \dots, i_k$, удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется возрастающей, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности $a_1, \dots, a_n$. Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел: $a_1, a_2, \dots, a_n$. Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

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

Задача D. Кошелёк

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

Рик превратил себя в кошелёк! Бум! Что вы на это скажете? Вот так поворот! Рик — кошелёк. Что вы на это скажете? Этого не происходило ещё никогда. Рик превратил себя в кошелёк! **КОШЕЛЁК-РИК!**

Так как Рик теперь не может двигаться, Морти вынужден помогать ему с обратным превращением. Для этого нужно правильным образом добавить в кошелёк несколько купюр. Всего есть n типов купюр, пронумерованных числами от 1 до n так, что чем больше номер купюры, тем больше ее номинал. В кошельке уже лежит некоторая сумма денег: для каждого i от 1 до n в кошельке уже лежит a_i купюр этого вида. Более того, купюры лежат там в ряд и отсортированы по номиналу, то есть если в кошельке лежат две купюры типов i и j , где $i < j$, то купюра типа i расположена левее купюры типа j . Это свойство должно сохраняться в каждый момент: по словам Рика, если его нарушить, последствия будут непредсказуемы!

Благо у Морти есть две изначально пустые стопки s_l и s_r , в которые он в процесс своих действий может класть купюры.

Кошелёк поддерживает 4 вида операций:

1. Вынуть из кошелька самую левую купюру и положить её с правого края стопки s_l .
2. Вынуть из кошелька самую правую купюру и положить её с левого края стопки s_r .
3. Вставить с левого края кошелька новую купюру.
4. Вставить с правого края кошелька новую купюру.

После каждой операции третьего или четвёртого типа необходимо тут же возвратить все вынутые купюры в кошелёк, а именно — все купюры стопки s_l , не меняя их расположения, поместить в левый край кошелька, а s_r — в правый край. В самом начале стопки s_l и s_r пустые.

Чтобы превратить Рика из кошелька назад в человека, Морти должен по очереди поместить в кошелёк m новых купюр, j -я добавленная купюра должна быть типа b_j . Купюры должны быть добавлены операциями 3-го и 4-го видов. После каждой операции купюры должны оставаться отсортированными по номиналу слева направо. Рик и Морти, чтобы понять, сколько времени у них займёт этот процесс, обратились к вам за помощью. Напишите программу, которая определит, какое наименьшее число операций первого и второго типов потребуется непосредственно перед каждым добавлением очередной купюры.

Формат входных данных

Первая строка входных данных содержит число n — количество различных типов купюр ($1 \leq n \leq 10^5$).

Вторая строчка содержит n целых чисел a_i — изначальное количество купюр каждого типа, которые лежат в кошельке ($0 \leq a_i \leq 10^5$).

В третьей строке находится число m — количество купюр, которые надо добавить в кошелёк, чтобы Рик обратно превратился в человека ($1 \leq m \leq 2 \cdot 10^5$).

В последней строке входных данных находится m чисел b_j — типы купюр, которые надо добавить в кошелёк, что бы Рик снова стал человеком ($1 \leq b_j \leq n$).

Формат выходных данных

В первой строке выходных данных выведите через пробел количество операций первого и второго типа, которое надо будет совершить перед каждым добавлением очередной купюры в кошелёк.

Пример

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

Замечание

Будем изображать текущее положение в формате s_l/s_r , где s — последовательность купюр, лежащих в кошельке.

В первом тесте в самом начале ситуация выглядит как $/1122233/$, после чего осуществляется 5 действий:

1. Чтобы добавить купюру типа 2, надо с любой из сторон дважды взять купюру, потом с той же стороны засунуть купюру типа 2.
2. В $/1122233/$ добавить купюру типа 1 можно, просто засунув эту купюру с левой стороны.
3. Из $/11122233/$ быстрее всего вынуть два раза купюру справа, получить $/111222/33$ и добавить справа купюру типа 2.
4. В $/111222233/$ сразу добавляем справа купюру типа 3.
5. Нет разницы, с какой стороны в $/1112222333/$ три раза вынуть купюру. Например, слева: получим $111/222333/$ и добавим слева купюру типа 2, получив финальную ситуацию $/1112222333/$.

Задача Е. Локи и Шахматы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	256 мегабайт

Однажды Тор заставил Локи играть в шахматы. Но шахматы в Асгарде не совсем обычные. Дело в том, что шахматная доска представляет собой таблицу из n строк и m столбцов. Каждая ячейка таблицы может либо иметь пешку, либо не иметь. Ячейка, которая не имеет пешки, называется свободной, а ячейка, которая имеет пешку, называется занятой.

Локи необходимо сделать q действий. Каждое действие заключается в том, что Тор дает Локи координаты ячейки и направление: 1 — вверх, 2 — вправо, 3 — вниз и 4 — влево. Если выбранная Тором ячейка свободна, то Локи ничего не должен делать, в противном случае он должен подвинуть пешку в указанном направлении.

Обратите внимание, что двигаются также все пешки, которые выбранная пешка толкает при движении в указанном направлении. Если пешка достигает границы доски, то дальше она не движется.

Формат входных данных

В первой строке входных данных содержатся три числа n , m и q ($1 \leq n, m \leq 1000, 1 \leq q \leq 10^6$). Далее в n строках содержится строка из m символов. $c_{i,j}$ — j -й символ в i -й строке. Если $c_{i,j}$ равно единице, то в данной ячейке стоит пешка, в противном случае ячейка является свободной. В последних q строках содержатся три целых числа x , y и dir — строка и столбец запроса и направление ($0 \leq x \leq n-1, 0 \leq y \leq m-1, 1 \leq dir \leq 4$). Строки нумеруются сверху вниз от 0 до $n-1$, столбцы нумеруются слева направо от 0 до $m-1$.

Формат выходных данных

В n строках выходных данных выведите m символов. В i -й строке j -й символ должен быть равен единице, если после выполнения всех команд от Тора в текущей ячейке находится пешка; в противном случае, если пешки в текущей ячейке нет, выведите 0.

Пример

стандартный ввод	стандартный вывод
3 3 6	000
000	010
010	000
000	
1 1 2	
1 1 2	
1 2 3	
2 2 2	
2 2 4	
2 1 1	

Задача F. Различные числа

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

Сколько различных чисел на отрезке массива?

Формат входных данных

В первой строке дано целое число n — длина массива ($1 \leq n \leq 300\,000$).

Во второй строке дано n целых чисел a_i — элементы массива ($0 \leq a_i < 10^9$).

В третьей строке дано целое число q — количество запросов ($1 \leq q \leq 300\,000$).

i -я из следующих q строк содержит пару целых чисел l_i, r_i — границы отрезка i -го запроса ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Выведите ответы на запросы по одному в строке.

Пример

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

Задача G. Ладьи и прямоугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

У Поликарпа есть шахматная доска размера $n \times m$, на которой расставлены k ладей. Поликарп еще не придумал правила игры, в которую он будет играть. Однако он уже выделил на доске q прямоугольных участков особой стратегической важности, которые должны быть надежно защищены. По мнению Поликарпа, прямоугольный участок доски надежно защищен, если все его свободные клетки бьются ладьями, стоящими на этом участке. Ладьи на остальной части доски на защиту участка не влияют. Расстановка ладей фиксирована и не может быть изменена. Напомним, что ладья бьет все клетки, расположенные с ней на одной вертикали или горизонтали, если между клеткой и ладьей нет других фигур. Помогите Поликарпу определить, все ли стратегически важные участки надежно защищены.

Формат входных данных

В первой строке содержатся четыре целых числа n , m , k и q ($1 \leq n, m \leq 100\,000$, $1 \leq k, q \leq 200\,000$) — размеры доски, количество ладей и количество стратегически важных участков. Будем считать, что клетки доски пронумерованы числами от 1 до n по горизонтали и от 1 до m по вертикали. Следующие k строк содержат пары целых чисел « $x\ y$ », описывающие положение ладей ($1 \leq x \leq n$, $1 \leq y \leq m$). Гарантируется, что все ладьи стоят в разных клетках. Следующие q строк описывают стратегически важные участки четверками чисел « $x_1\ y_1\ x_2\ y_2$ » ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq m$). Соответствующий прямоугольный участок состоит из клеток (x, y) , для которых $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$. Стратегически важные участки могут пересекаться или совпадать.

Формат выходных данных

Выведите q строк. Для каждого стратегически важного участка выведите «YES», если он надежно защищен, и «NO» в противном случае.

Примеры

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

Задача Н. Вика и отрезки

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

У Вики есть бесконечный лист клетчатой бумаги. Изначально каждая клетка белого цвета. Она ввела на этом листе двумерную систему координат и нарисовала n чёрных горизонтальных и вертикальных отрезков, параллельных осям координат и шириной в одну клетку. Таким образом, каждый отрезок является множеством соседних клеток, расположенных в одной строке или в одном столбце.

Перед вами стоит задача посчитать число закрашенных клеток. Если клетка была покрашена более одного раза, в ответе она должна быть посчитана ровно один раз.

Формат входных данных

В первой строке входных данных следует целое положительное число n ($1 \leq n \leq 100\,000$) – количество отрезков, нарисованных Викой.

В следующих n строках заданы по четыре целых числа $x1, y1, x2, y2$ ($-10^9 \leq x1, y1, x2, y2 \leq 10^9$) – координаты концов отрезков, нарисованных Викой. Гарантируется, что все отрезки параллельны осям координат. Отрезки могут пересекаться, накладываться друг на друга и даже полностью совпадать.

Формат выходных данных

Выведите количество закрашенных Викой клеток. Если клетка была покрашена более одного раза, то в ответе она должна быть посчитана ровно один раз.

Примеры

стандартный ввод	стандартный вывод
3 0 1 2 1 1 4 1 2 0 3 2 3	8
4 -2 -1 2 -1 2 1 -2 1 -1 -2 -1 2 1 2 1 -2	16

Задача I. Поиск пирамиды

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

«Фуфелшмертц Пакость Инкорпорейтед» опять пакостит! Теперь он ежедневно сдвигает литосферные плиты Земли. Перри-утконос получил важное задание: каждый день искать самый подозрительный рельеф на прямой и затем, разумеется, сообщать о нем в агентство.

У него под наблюдением находятся n участков, расположенных на одной прямой. Каждый участок характеризуется одним числом h_i — высотой данного участка над уровнем моря. Отрезок называется *подозрительным*, если на нем существует такой участок, что высоты участков левее него строго возрастают, а правее — строго убывают. При этом, из-за проделок Фуфелшмерца высоты участков постоянно меняются.

Помогите Перри определить длину самого длинного подозрительного отрезка участков после каждого изменения. Гарантируется, что в любой момент времени нет двух рядом стоящих участков с одинаковой высотой.

Формат входных данных

В первой строке дано одно число n — количество участков ($1 \leq n \leq 100\,000$). Во второй строке дано n чисел — высоты участков ($|h_i| \leq 10^{18}$).

В третьей строке дано число m — количество изменений ($1 \leq m \leq 100\,000$). В следующих m строках дано по два целых числа x и y — индекс участка, высота которого изменилась, и новое значение высоты для этого участка, соответственно ($1 \leq x \leq n$, $|y| \leq 10^{18}$).

Формат выходных данных

Выведите m чисел, i -е из которых равно длине наибольшего подозрительного отрезка после i -го изменения.

Пример

стандартный ввод	стандартный вывод
9	6
1 2 3 4 5 4 3 2 1	6
5	4
3 10	5
2 5	5
7 100000000	
5 1	
3 1	

Задача J. Сад пермского периода

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Оранжерея «Сад пермского периода» представляет собой прямоугольный участок для выращивания растений пермского периода. Оранжерея была разбита дорожками на квадраты. В центре каждого квадрата посажено одно растение. Размер квадрата зависит от корневой системы растения.

За год дорожки заросли травой, что затруднило уход за оранжереей. Чтобы при садовых работах не повредить корневую систему какого-либо растения, по имеющемуся расположению растений необходимо восстановить размеры соответствующих им квадратов.

Введем декартову прямоугольную систему координат, начало которой совмещено с левым нижним углом оранжереи. Ось Ox направлена вдоль нижней границы участка, ось Oy – вдоль левой. Изначально дорожки были проложены параллельно осям координат. Единичный отрезок удалось выбрать так, что координаты углов каждого из квадратов оказались целыми.

Требуется написать программу, которая по размеру оранжереи и координатам растений определит размеры соответствующих им квадратов.

Формат входных данных

В первой строке входного файла записаны три натуральных числа: W – ширина оранжереи, H – длина оранжереи и N – количество посаженных растений. В каждой из следующих N строк расположены по два числа: x_i, y_i – координаты i -го растения ($0 < x_i < W, 0 < y_i < H$). Гарантируется, что соответствующие растениям квадраты имеют целую длину стороны и покрывают всю оранжерею.

$$N \leq 2 \cdot 10^5$$
$$\max(W, H) \leq 10^{12}$$

Формат выходных данных

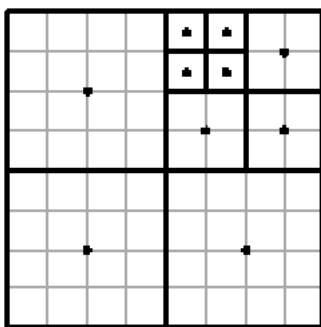
В выходной файл необходимо вывести N целых чисел – размеры квадратов, соответствующих растениям. Числа требуется вывести в порядке описания растений во входном файле.

Примеры

стандартный ввод	стандартный вывод
4 6 3 1 1 3 1 2 4	2 2 4
8 8 10 4.5 7.5 5.5 7.5 2 6 4.5 6.5 7 7 5 5 6 2 7 5 2 2 5.5 6.5	1 1 4 1 2 2 4 2 4 1
7 7 1 3.5 3.5	7
8 8 1 4 4	8

Замечание

Оранжерея во втором примере соответствует следующему рисунку:



Задача К. Приемная короля Кога

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

Король Ког устал от обычной небрежности своих рыцарей — они могут врываться в его зал без предварительного уведомления! Поэтому король решил построить приемную с очередью, где каждый рыцарь заранее выбирает время своего прихода и продолжительность визита. Рыцари обслуживаются в порядке записанного времени, но каждый рыцарь должен ждать, пока визиты всех рыцарей перед ним не завершатся.

Принцесса Кеабини хочет увидеть своего отца. Однако она не хочет прерывать рыцарей, поэтому она становится в очередь. К сожалению, рыцари очень часто меняют свои решения — они могут присоединяться к очереди или отменять свои визиты. Пожалуйста, помогите принцессе понять, сколько времени ей придется ждать, пока она увидит своего отца, если она войдет в очередь в указанные моменты времени, учитывая записи в приемной.

Формат входных данных

Первая строка входных данных содержит одно целое число q ($1 \leq q \leq 3 \cdot 10^5$) — количество событий. Событие может быть трех типов: присоединение, отмена или запрос.

- Присоединение “+ t d ” ($1 \leq t, d \leq 10^6$) — новый рыцарь присоединяется к очереди, где t это время, когда рыцарь придет, а d это продолжительность визита.
- Отмена “- i ” ($1 \leq i \leq q$) — рыцарь отменяет визит, где i это номер (считается с единицы) соответствующего события присоединения в списке всех событий.
- Запрос “? t ” ($1 \leq t \leq 10^6$) — Кеабини спрашивает, сколько времени ей придется ждать, если она придет в момент времени t .

Гарантируется, что после каждого события в очереди нет двух рыцарей с одинаковым временем прихода. События отмены относятся к предыдущим присоединениям, которые еще не были отменены.

Кеабини может прийти в то же время, что и какой-либо рыцарь, но Кеабини очень вежлива и она будет ждать, пока рыцарь пройдет.

Формат выходных данных

Для каждого запроса напишите отдельную строку с количеством времени, которое Кеабини придется ждать.

Пример

стандартный ввод	стандартный вывод
19	0
? 3	1
+ 2 2	0
? 3	2
? 4	1
+ 5 2	3
? 5	2
? 6	1
+ 1 2	2
? 2	1
? 3	0
? 4	0
? 5	2
? 6	1
? 7	1
? 9	
- 8	
? 2	
? 3	
? 6	

Задача L. Разделить ИЛИ Завоевать

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

Вам дан массив $[a_1, a_2, \dots, a_n]$, состоящий из целых чисел от 0 до 10^9 . Вам нужно разделить этот массив на несколько сегментов (возможно, один) таким образом, чтобы каждый элемент принадлежал ровно одному сегменту.

Пусть первый сегмент будет массивом $[a_1, a_{l_1+1}, \dots, a_{r_1}]$, второй сегмент — $[a_{l_2}, a_{l_2+1}, \dots, a_{r_2}]$, ..., последний сегмент — $[a_{l_k}, a_{l_k+1}, \dots, a_{r_k}]$. Поскольку каждый элемент должен принадлежать ровно одному массиву, $l_1 = 1$, $r_k = n$, и $r_i + 1 = l_{i+1}$ для каждого i от 1 до $k - 1$. Разделение должно удовлетворять следующему условию:

$$f([a_{l_1}, a_{l_1+1}, \dots, a_{r_1}]) \leq f([a_{l_2}, a_{l_2+1}, \dots, a_{r_2}]) \leq \dots \leq f([a_{l_k}, a_{l_k+1}, \dots, a_{r_k}]),$$

где $f(a)$ — это побитовая ИЛИ всех элементов массива a .

Вычислите количество способов разделить массив и выведите его по модулю 998 244 353. Два способа считаются различными, если последовательности $[l_1, r_1, l_2, r_2, \dots, l_k, r_k]$, обозначающие разделения, различны.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10^5$) — длина массива a .

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^6$) — элементы данного массива.

Формат выходных данных

Выведите одно целое число — количество способов разделить массив, взятое по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	4
5 1000 1000 1000 1000 1000	16
3 3 4 6	3

Замечание

В первых двух примерах каждый способ разделить массив является допустимым.

В третьем примере есть три допустимых способа разделить массив:

- $k = 3$; $l_1 = 1, r_1 = 1$, $l_2 = 2, r_2 = 2$, $l_3 = 3, r_3 = 3$; полученные массивы — $[3]$, $[4]$, $[6]$, и $3 \leq 4 \leq 6$;
- $k = 2$; $l_1 = 1, r_1 = 1$, $l_2 = 2, r_2 = 3$; полученные массивы — $[3]$ и $[4, 6]$, и $3 \leq 6$;
- $k = 1$; $l_1 = 1, r_1 = 3$; будет только один массив: $[3, 4, 6]$.

Если вы разделите массив на два массива $[3, 4]$ и $[6]$, побитовая ИЛИ первого массива равна 7, а побитовая ИЛИ второго массива равна 6; $7 > 6$, поэтому этот способ разделить массив недопустим.