



Общая информация по задачам олимпиады

Ограничение по памяти

Во всех задачах ограничение составляет 512 МБ.

Ограничение на размер исходного кода программы

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Ограничение на посылку решений

По каждой задаче на проверку принимается не более 50 решений.

По каждой задаче участник не может отправить решение более одного раза в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

Система оценки

Каждая задача олимпиады поделена на несколько подзадач. Чтобы набрать баллы по подзадаче, программа должна пройти все тесты этой подзадачи.

За каждую задачу выставляется суммарный балл по всем ее подзадачам. В каждой подзадаче оценивается лучшее решение, то есть за подзадачу выставляется максимальный набранный по ней балл среди всех решений.

Получение информации о результатах проверки

Чтобы получить информацию о проверке вашего решения, используйте ссылку «Информация о проверке» во вкладке «Решения» в PCMS2 Web Client. По каждой задаче вам будет доступна информация по количеству набранных баллов в каждой подзадаче или результат проверки на первом непройденном тесте.

Таблица результатов

Во время соревнования доступна текущая таблица результатов. Для доступа к ней используйте ссылку «Результаты» в PCMS2 Web Client. Таблица результатов в PCMS2 Web Client не является окончательной.



Задача A. Universal Paperclips

Ограничение по времени: 2 секунды

Universal Paperclips — инкрементальная игра, целью которой является изготовление скрепок. В вашем распоряжении есть оборудование, производящее скрепки по нажатию кнопки.

В упрощенной версии игры для этой задачи будем считать, что игроку доступны два вида действий:

1. Нажать на кнопку. Каждое нажатие добавляет определенное количество скрепок. Обозначается буквой «C» («click»).
2. Улучшить процесс изготовления. Благодаря этому, каждое последующее нажатие на кнопку будет добавлять на одну скрепку больше. Обозначается буквой «U» («upgrade»).

Изначально у игрока ноль скрепок, а каждое нажатие на кнопку добавляет одну скрепку. На каждое улучшение тратится x скрепок. **Во всех входных данных гарантируется, что перед каждым улучшением в наличии есть хотя бы x скрепок.**

Бенни решил написать бота для этой игры. Он закодировал в бота последовательность из n действий S , которую бот выполняет по циклу. Каждую секунду бот совершает очередное действие: на 1-й секунде бот применяет S_1 , на второй — S_2 , ..., на n -й — S_n , на $n + 1$ -й — снова S_1 , и далее по циклу.

Сколько скрепок будет у Бенни после t секунд работы его бота?

Формат входных данных

В первой строке содержатся три числа n , t и x ($1 \leq n \leq 10^5$; $1 \leq t \leq 10^9$; $0 \leq x < n$). Во второй строке содержится строка S длины n из символов «C» и «U».

Формат выходных данных

Выведите одно число — число скрепок через t секунд.

Система оценки

Подзадача	Баллы	Ограничения
1	33	$t \leq 10^5$
2	67	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
10 15 2 CCCCUCCU	25
3 10 0 UUU	0
9 150 1 CCUCCCCU	2023

Замечание

Рассмотрим первый пример. Первые пять операций приносят по одной скрепке каждая. Следующие две операции используют по две скрепки, после чего остается одна скрепка, однако, следующие два нажатия производят по три скрепки. Далее, улучшение снова отбирает две. Последние пять нажатий приносят по 4 скрепки каждое.

Итоговое число скрепок равно $1 + 1 + 1 + 1 + 1 - 2 - 2 + 3 + 3 - 2 + 4 + 4 + 4 + 4 + 4 = 25$.



Задача В. Hanoi Chips

Ограничение по времени: 2 секунды

В городе Ханой помимо известной башни Брама также существует и числовая прямая Брама. На этой прямой находятся три абсолютно одинаковые фишки, которые можно по ней переставлять. За один ход можно взять фишку и переставить ее в новую позицию, которая симметрична старой относительно одной из двух других фишек. К примеру, если фишки лежат в координатах 1, 4 и 6, то фишку на позиции 4 можно переставить в позицию -2 (симметрично относительно первой фишки), либо в позицию 8 (симметрично относительно третьей фишки). Одновременно в одной точке может находиться несколько фишек.

По одной из легенд, как только фишки займут определенные позиции на прямой, с неба посыплются самые настоящие сокровища. Никто не знает точные координаты, в которых должны оказаться фишки, но у вас есть одно предположение.

По начальным и конечным координатам фишек проверьте, можно ли получить конечное положение, соблюдая правила, и если можно, найдите подходящий порядок перемещений. Минимизировать количество перемещений не обязательно, но разрешается сделать не более 10^5 перемещений. Промежуточные координаты фишек не могут превышать 10^9 по модулю.

Формат входных данных

Первая строка входных данных содержит три целых числа x_1, x_2, x_3 ($-10^4 \leq x_i \leq 10^4$) — изначальные координаты фишек.

Вторая строка входных данных содержит три целых числа y_1, y_2, y_3 ($-10^4 \leq y_i \leq 10^4$) — конечные координаты фишек, которые нужно получить.

Формат выходных данных

Если перевести фишки в конечное положение невозможно, выведите в первой строке единственное число «-1» (без кавычек).

Иначе, выведите в первой строке единственное число k ($0 \leq k \leq 10^5$) — количество перемещений фишек. Далее в каждой следующей строчке выведите очередное перемещение фишек в следующем формате: первое число означает изначальную координату фишки, которую нужно переставить, а второе — конечную. Все числа не должны превосходить 10^9 по модулю.

Система оценки

Подзадача	Баллы	Ограничения
1	17	Как в начальном, так и в конечном положении, фишки занимают три идущие подряд позиции (например, 3, 4, 5 или $-7, -9, -8$)
2	20	$0 \leq x_i, y_i \leq 50$
3	25	Конечное положение можно получить за ≤ 8 действий
4	38	Без дополнительных ограничений



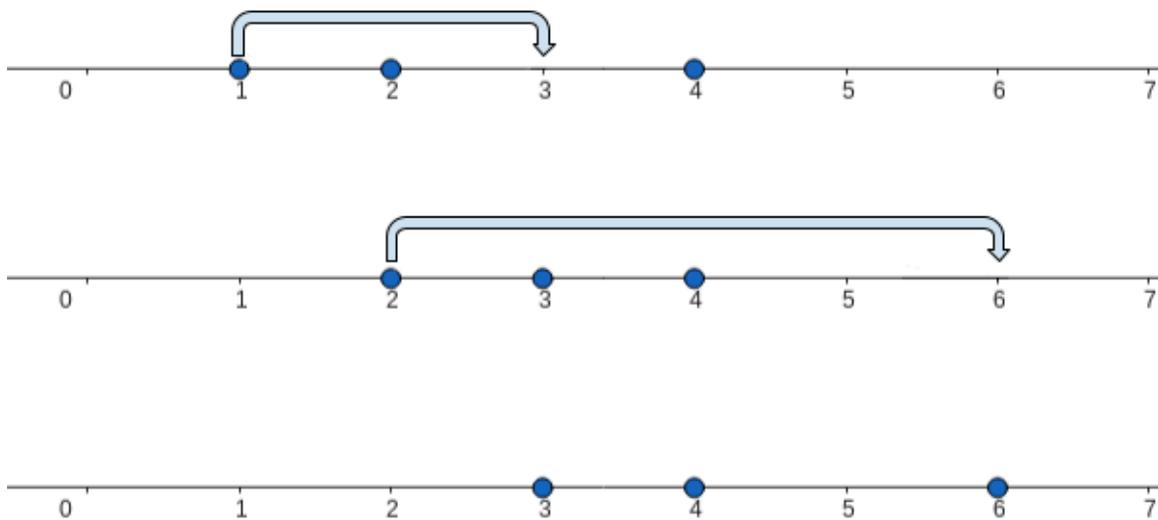
Примеры

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

Замечание

Обратите внимание, что поскольку фишки абсолютно одинаковые, фишки не обязательно должны перейти в конце именно в соответствующие координаты. Например, x_1 может перейти в y_2 , x_2 в y_3 и x_3 в y_1 .

В первом примере нужно перевести фишки из координат (1, 2, 4) в координаты (3, 4, 6). Для этого можно сделать два перемещения: сначала переставить фишку с координатой 1 в точку с координатой 3, а потом фишку с координатой 2 в точку с координатой 6.





Задача C. Sorting Subarrays

Ограничение по времени: 2 секунды

Вам дан массив целых чисел. Вы должны ровно один раз применить к этому массиву операцию сортировки подотрезка, то есть выбрать какой-то непустой подотрезок массива и переупорядочить элементы этого подотрезка в неубывающем порядке.

Сколько различных массивов вы можете получить в результате такой операции? Подотрезком массива называются несколько подряд идущих элементов массива.

Формат входных данных

Первая строка содержит число n — размер массива ($1 \leq n \leq 200\,000$).

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n — элементы массива ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите количество различных массивов, которое можно получить после применения одной операции сортировки подотрезка.

Система оценки

Подзадача	Баллы	Ограничения
1	13	$n \leq 50$
2	26	$n \leq 3\,000$
3	22	$1 \leq a_i \leq 2$
4	39	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
7 3 1 4 1 5 9 2	8
4 1 2 3 4	1

Замечание

Ниже перечислены все возможные различные массивы, которые можно получить в первом примере, и один из способов получить каждый из них. Сортируемые подотрезки отмечены квадратными скобками.

- $[3], 1, 4, 1, 5, 9, 2 \rightarrow 3, 1, 4, 1, 5, 9, 2$
- $[3, 1], 4, 1, 5, 9, 2 \rightarrow 1, 3, 4, 1, 5, 9, 2$
- $[3, 1, 4, 1], 5, 9, 2 \rightarrow 1, 1, 3, 4, 5, 9, 2$
- $[3, 1, 4, 1, 5, 9, 2] \rightarrow 1, 1, 2, 3, 4, 5, 9$
- $3, [1, 4, 1], 5, 9, 2 \rightarrow 3, 1, 1, 4, 5, 9, 2$
- $3, [1, 4, 1, 5, 9, 2] \rightarrow 3, 1, 1, 2, 4, 5, 9$
- $3, 1, 4, [1, 5, 9, 2] \rightarrow 3, 1, 4, 1, 2, 5, 9$
- $3, 1, 4, 1, 5, [9, 2] \rightarrow 3, 1, 4, 1, 5, 2, 9$



Задача D. RestORe

Ограничение по времени: 2 секунды

Рассмотрим последовательность целых неотрицательных чисел от L до R включительно. Разобьем такую последовательность на n непустых отрезков и посчитаем побитовое «или» чисел в каждом отрезке. Результат для i -го отрезка назовем f_i .

Решите обратную задачу: по числам f_1, f_2, \dots, f_n найдите количество способов выбрать L и R , а затем разбить все числа от L до R на n непустых отрезков так, чтобы результаты вычисления побитового «или» в каждом отрезке были равны f_1, f_2, \dots, f_n соответственно.

Побитовое «или» (or) двух целых неотрицательных чисел вычисляется следующим образом: запишем оба числа в двоичной системе счисления, i -й двоичный разряд результата равен 1, если хотя бы одного из аргументов он равен 1. Например, $(17 \mid 13) = (10001_2 \mid 01101_2) = 11101_2 = 29$.

Формат входных данных

В первой строке содержится число n ($1 \leq n \leq 200\,000$). В следующей строке содержатся n чисел f_i ($0 \leq f_i < 2^{60}$).

Формат выходных данных

Выведите одно число — количество способов выбрать последовательность из подряд идущих чисел и разбить ее на n отрезков. Поскольку ответ может быть большим, выведите его по модулю 1 000 000 007.

Система оценки

Подзадача	Баллы	Ограничения
1	16	$n = 1$
2	12	$n \leq 16, 0 \leq f_i < 16$
3	13	$n \leq 256, 0 \leq f_i < 256$
4	18	$n \leq 1024, 0 \leq f_i < 1024$
5	24	$n \leq 30\,000, 0 \leq f_i < 2^{30}$
6	17	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
3 0 7 7	4
1 11	6
2 4 6	0

Замечание

В первом примере существует четыре способа разбить массив подряд идущих чисел на три части, так, чтобы побитовые «или» в них были равны 0, 7 и 7 соответственно:

- $[0], [1, 2, 3, 4, 5, 6], [7]: 0 = 0; 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 = 7; 7 = 7$
- $[0], [1, 2, 3, 4, 5], [6, 7]: 0 = 0; 1 \mid 2 \mid 3 \mid 4 \mid 5 = 7; 6 \mid 7 = 7$
- $[0], [1, 2, 3, 4], [5, 6, 7]: 0 = 0; 1 \mid 2 \mid 3 \mid 4 = 7; 5 \mid 6 \mid 7 = 7$
- $[0], [1, 2, 3, 4], [5, 6]: 0 = 0; 1 \mid 2 \mid 3 \mid 4 = 7; 5 \mid 6 = 7$



Задача E. Non-adjacent Swaps

Ограничение по времени: 2 секунды

Дана перестановка. За одну операцию в ней можно поменять местами два соседних элемента, значения которых отличаются больше чем на один.

Рассмотрим все перестановки, которые можно получить из данной этими операциями. Назовем *расстоянием* между двумя перестановки минимальное число таких операций, необходимых для того, чтобы превратить первую перестановку во вторую.

Найдите их количество, а также сумму расстояний от начальной перестановки до всех достижимых.

Формат входных данных

В первой строке содержится число n ($1 \leq n \leq 100$) — число элементов перестановки. Вторая строка содержит n различных чисел a_i ($1 \leq a_i \leq n$) — начальную перестановку.

Формат выходных данных

Выведите два числа — количество перестановок, которые можно получить, а также сумму расстояний от начальной до всех достижимых. Выведите оба числа по модулю 1 000 000 007.

Система оценки

Подзадача	Баллы	Ограничения
1	15	$n \leq 8$
2	20	$n \leq 15$
3	30	$n \leq 30$
4	20	$n \leq 50$
5	15	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
4 3 1 4 2	5 5
5 1 2 3 4 5	1 0
6 5 3 1 2 4 6	61 218

Замечание

В первом примере из начальной перестановки можно получить следующие:

- [3, 1, 4, 2], за 0 операций;
- [3, 1, 2, 4], за 1 операцию;
- [1, 3, 4, 2], за 1 операцию;
- [3, 4, 1, 2], за 1 операцию;
- [1, 3, 2, 4], за 2 операции (меняются местами элементы со значениями (3, 1) и (4, 2)).

Всего пять перестановок, сумма расстояний до них также равна пяти.

Во втором примере никакие два элемента нельзя поменять местами, поэтому достижима только одна перестановка с расстоянием 0.