

Задача А. Миша и математика

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

Миша сидел на занятиях математики в Высшей школе экономики и решал следующую задачу: дано n целых чисел и нужно расставить между ними знаки $+$ и \times так, чтобы результат полученного арифметического выражения был нечётным (например, между числами 5, 7, 2, можно расставить арифметические знаки следующим образом: $5 \times 7 + 2 = 37$). Так как примеры становились все больше и больше, а Миша срочно убегает в гости, от вас требуется написать программу решающую данную задачу.

Формат входных данных

В первой строке содержится единственное число n ($2 \leq n \leq 10^5$). Во второй строке содержится n целых чисел a_i , разделённых пробелами ($-10^9 \leq a_i \leq 10^9$). Гарантируется, что решение существует.

Формат выходных данных

В одной строке выведите $n - 1$ символ $+$ или \times , в результате применения которых получается нечётный результат. (Для вывода используйте соответственно знаки «+» (ASCII код—43) и «x» (ASCII код—120), без кавычек).

Система оценки

Решения верно работающие при $n \leq 10$ будут набирать не менее 40% баллов.

Примеры

стандартный ввод	стандартный вывод
3 5 7 2	x+
2 4 -5	+

Задача В. Очередь за комплексом

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

Федок очень хочет купить себе комплексный обед в столовой, но сделать это не так просто. В столовой работает всего одна касса, и то очень медленно. На данный момент в очереди находится n ($2 \leq n \leq 100\,000$) людей, а сам Федок находится на k -ой ($1 \leq k \leq n$) позиции в ней. И вот, чтобы занять себя в этой очереди, Федок стал обдумывать коварный план как побыстрее оплатить комплексный обед и начать есть.

В чем состоит коварный план? Федок хочет крикнуть, что открылась новая касса. Тогда, по его мнению, многие уйдут из очереди в поисках этой кассы, а он приблизится к заветной еде. Но вот в чем проблема: не все люди так нетерпеливы как наш герой. Проще говоря, у каждого человека есть свой параметр P_i — терпеливость. Если человек стоит на позиции x и его терпеливость равна y , то он уйдет искать новую кассу только в том случае, если $y < x$.

Казалось бы, эта задача трудна, но и Федок не глуп. Он своим метким взором определил терпеливости всех людей, стоящих в очереди кроме него. Увы, так как людей очень много, в его голове все перепуталось, и некоторые числа поменялись местами. Таким образом наш герой получил некоторую перестановку множества терпеливости всех людей в очереди P_1, P_2, \dots, P_{n-1} .

Федок не знает точно, кто насколько терпелив и боится, что не сдвинется в очереди после реализации своей задумки. Поэтому он просит вас помочь ему узнать, какую минимальную и максимальную позицию от начала очереди он может занимать после того как крикнет: «Свободная касса!»

Формат входных данных

В первой строке входных данных находится число людей в очереди n ($2 \leq n \leq 100\,000$)

Во второй строке находится число k ($1 \leq k \leq n$) — текущая позиция Федка в очереди

В следующей строке содержатся $n - 1$ число — перестановка множества терпеливостей людей в очереди ($0 \leq P_i \leq n$)

Формат выходных данных

В первой строке выведите минимальное место, которое может стать у Федка после применения его плана

Во второй строке выведите максимальное место, которое может стать у Федка после применения его плана

Система оценки

- Решения, работающие для $n \leq 10$ будут набирать не менее 5 баллов
- Решения, работающие для $n \leq 1000$ будут набирать не менее 10 баллов

Пример

стандартный ввод	стандартный вывод
4	1
2	2
0 3 3	

Замечание

Пояснение к первому тесту:

Федок стоит на второй позиции. Перед ним один человек.

Если его терпеливость будет равна нулю, то он уйдет, и Федок станет первым

Если же его терпеливость равна трем, то он не уйдет, и Федок останется второй

Задача С. Игра после сессии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

После того как Данил сдал всю сессию на 10 из 10, он решил расслабиться и поиграть в компьютерные игры. Его ноутбук был очень старым, поэтому Данил смог запустить только одну игру, в которой паркуррист Похпид прыгает по крышам небоскребов.

На каждом уровне игры генерируется новый проспект из небоскребов, где каждая высотка задается уникальной положительной координатой относительно начала проспекта. После этого Похпид выбирает здание, с которого он начинает свой путь, при этом он может прыгать только на здание, координата которого больше текущей. Трудность этой игры заключается в том, что после каждого прыжка Похпид устает и уже не может прыгать также далеко как раньше. Формально говоря, на каждом уровне генерируется n зданий, координаты которых равны a_1, a_2, \dots, a_n . Игрок выбирает стартовый небоскреб и с него может прыгнуть на любое здание, которое находится правее, при этом длина первого прыжка может быть любой. Для всех последующих прыжков должно быть выполнено условие: если сейчас игрок находится на здании с координатой a_i , а до этого был на позиции a_j , то он может перепрыгнуть на здание с координатой a_k , если $a_i - a_j > a_k - a_i$.

Так как Данил любит делать все по максимуму, он решил на каждом уровне посетить максимально возможное количество зданий. Помогите ему с этой задачей и для каждого уровня найдите максимальное количество небоскребов, которое можно посетить.

Формат входных данных

В первой строке входных данных дается число уровней t ($1 \leq t \leq 1000$).

В следующих строках каждый уровень задается числом зданий n ($1 \leq n \leq 5000$) в одной строке и позициями этих зданий a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^{18}$, $a_i \neq a_j$ если $i \neq j$) в следующей строке. Гарантируется, что сумма n по всем уровням не превосходит 30 000.

Формат выходных данных

Для каждого уровня нужно вывести в отдельной строке максимальное количество небоскребов, которое может посетить Данил.

Система оценки

Решения, работающие при $n \leq 10$ и $t = 1$ будут получать не меньше 25% баллов.

Решения, работающие при $n \leq 100$ и $t \leq 10$ будут получать не меньше 50% баллов.

Решения, работающие при $n \leq 2500$ и $t \leq 10$ будут получать не меньше 75% баллов.

Примеры

стандартный ввод	стандартный вывод
1 5 9 19 3 8 6	4
3 3 16 5 17 8 3 8 2 10 15 11 14 12 6 12 19 11 17 2 5	3 5 4

Замечание

В первом примере оптимальный выбор небоскребов 3, 5, 4, 1 их координаты соответственно будут равны 3, 6, 8, 9.

Во втором примере ответы для уровней получаются выбором следующих 3 наборов индексов соответственно:

1) 2, 1, 3

2) 3, 2, 8, 7, 5

3) 5, 1, 4, 2

Задача D. Дек

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

Есть шарики пронумерованные от 1 до n , расположенные в виде вертикальной линии, которую мы назовем деком. Поступает q запросов одного из двух типов:

- *add x* - добавить в дек шарик с номером x . Вы можете положить шарик либо сверху, либо снизу дека.
- *del x* - удалить из дека шарик с номером x . Вы находите шарик с номером x в деке, достаете все шарики НИЖЕ x , удаляете шарик x , потом кладете достанные шарики (без x) обратно в том же порядке.

На запросы 1-го типа вы тратите 1 действие, а на запросы 2-го типа - $2k + 1$ действий, где k - количество шариков, которые лежат ниже удаляемого (то есть вы сначала достаете k шариков, потом удаляете нижний, потом кладете достанные k шариков обратно в том же порядке).

Посчитайте, какое минимальное количество действий вы можете потратить.

Формат входных данных

Первая строка содержит числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) - максимальный номер шарика и количество запросов. Далее следуют q строк в формате:

- *add x* - добавить шарик с номером x ($1 \leq x \leq n$) в начало или конец дека.
- *del x* - удалить шарик с номером x ($1 \leq x \leq n$) из дека.

Гарантируется, что никакой шарик не добавляется 2 раза, а также, что если есть запрос удаления шарика x , то он присутствует в деке в данный момент.

Формат выходных данных

Выведите единственное число - минимальное количество операций, которое вы можете потратить на запросы.

Система оценки

Решения, работающие при $q \leq 20$ будут получать не менее 15% баллов.

Решения, работающие в случае, если в любой момент времени в деке хранится не больше 6 элементов, $q \leq 1000$ будут набирать не менее 30% баллов.

Решения, работающие в случае, если в любой момент времени в деке хранится не больше 10 элементов, $q \leq 1000$ будут набирать не менее 40% баллов.

Решения, работающие при $q \cdot n \leq 10^6$ будут набирать не менее 65% баллов.

Решения, работающие при $q \leq 5000$ будут набирать не менее 70% баллов.

Примеры

стандартный ввод	стандартный вывод
2 4 add 1 del 1 add 2 del 2	4
8 6 add 5 add 8 del 5 add 1 add 4 del 1	6

Замечание

В первом примере независимо от того, как вы добавите шарики, вы потратите 4 действия (на удаление и на добавление каждого из шариков).

Во втором примере вам необходимо добавить шарик 5 вниз, шарик 8 вверх, шарик 1 вниз, шарик 4 вверх, тогда вы потратите ровно 6 действий.

Задача Е. Глеб и медиана

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

Глеб устал от побитового исключающего «или» и решил, что пора найти новую интересную функцию. Его выбор пал на медиану. Напомним, медианой массива называется число, которое окажется посередине, если массив упорядочить по возрастанию. В рамках этой задачи для массивов чётной длины положим медиану равной левому из двух центральных в отсортированном порядке элементов.

Для некоторого числа m назовём m -разбиением массива такое его разбиение на непересекающиеся отрезки, что на каждом из этих отрезков медиана больше либо равна m . Вам дан массив a длины n и q запросов двух видов:

1. присвоить элементу с индексом i значение x ;
2. найти наибольшее число k такое, что для подотрезка массива с индексами от l до r включительно существует m -разбиение на k отрезков.

Формат входных данных

В первой строке дается число n ($1 \leq n \leq 2 \cdot 10^5$) - размер массива. В следующей строке вводятся n чисел a_i ($1 \leq a_i \leq 10^9$) - элементы массива, на следующей строке вводится число q ($1 \leq q \leq 2 \cdot 10^5$) - количество запросов. В следующих q строках даются запросы, каждый в одном из следующих форматов:

- 1 i x — запрос 1 типа ($1 \leq i \leq n, 1 \leq x \leq 10^9$);
- 2 m l r — запрос 2 типа ($1 \leq m \leq 10^9, 1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждого запроса второго типа в отдельной строке выведите ответ на запрос. В случае если для отрезка не существует никакого m -разбиения, выведите 0.

Система оценки

В задаче присутствуют 6 групп:

1. $n \leq 5, q \leq 5$, такие решения будут набирать не менее 10% баллов
2. $n \leq 100, q \leq 100$, такие решения будут набирать не менее 20% баллов
3. $n \leq 10000, q \leq 10000$, такие решения будут набирать не менее 30% баллов
4. m - одно и то же для всех запросов, такие решения будут набирать не менее 25% баллов
5. нет запросов изменения, такие решения будут набирать не менее 35% баллов
6. Ограничения, как и в задаче

Пример

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