

Задача А. Мёд для Михаила

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

Медведь Михаил очень любит мёд. Вот беда — мёд у него в берлоге закончился, поэтому Михаил приехал к любимой бабушке, владеющей медовой фабрикой.

Бабушка очень добрая, она разрешила Мише брать столько мёда, сколько он хочет. Всего у бабушки есть n бочек с мёдом. В i -й бочке содержится a_i литров мёда. Миша взял с собой m ведер, каждое из которых вмещает p литров мёда.

Михаил очень хорошо разбирается в мёде и понимает, что если смешивать мёд из разных бочек, то он будет невкусный. Поэтому Миша не будет наливать мёд из нескольких бочек в одно ведро. Теперь Миша хочет узнать, какое максимальное количество мёда он сможет увезти с собой.

Формат входных данных

Первая строка содержит три целых числа n , m и p ($1 \leq n, m \leq 10^5$, $1 \leq p \leq 10^9$) — количество бочек с мёдом, количество ведер у Михаила и вместимость ведер, соответственно.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_i ($1 \leq a_i \leq 10^9$) — количество литров мёда в каждой из бочек.

Формат выходных данных

Выведите одно число — максимальное количество литров мёда, которое сможет унести Михаил.

Примеры

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

Замечание

В первом примере Миша нальет 3 литра мёда из второй бочки в одно ведро и 4 литра из третьей бочки во второе ведро.

Во втором примере Мише следует налить 3 и 4 литра мёда в два ведра из третьей бочки.

Задача В. База отдыха

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

База отдыха приглашает на летние каникулы классы школ размещение в прибрежных домиках с видом на море. Всего на базе отдыха N домиков. Автоматическая система бронирования позволяет производить определенные операции управления размещением в домиках автоматически. Команды системы приведены ниже:

Order Name M — забронировать M домиков для группы детей с названием *Name*. Гарантируется, что все названия групп уникальны. В ответ на эту команду система выводит номера домиков, которые система забронировала для этой группы детей. При этом система обязана выделить группе школьников как можно ближе расположенные друг к другу свободные домики, а первый из них — это первый свободный на побережье дом. Номера домиков разделены пробелом.

Cancel Name — отмена заказа для группы детей *Name*. После этой команды те номера домов, в которых предполагалась проживание группы школьников становятся свободными для бронирования. Гарантируется, что команды на отмену заказов поступают в систему в точно таком же порядке, в котором они поступали для бронирования.

Период бронирования начинается сегодня и все номера домиков свободны. Ожидается подача K запросов. Система должна работать бесперебойно, обеспечивая быстрые ответы на вводимые команды. В конце работы программы нужно вывести номера всех свободных домиков, которые еще можно предложить школьникам после обработки всех запросов.

Формат входных данных

В первой строке вводятся целые числа N и K — изначальное количество домиков, а также количество запросов ($1 \leq N, K \leq 10^5$).

Затем следуют K строк — сами запросы.

Если сейчас подается запрос первого типа, то вводится **Order Name M**, где *Name* — название группы детей, а M — количество домиков, которое надо забронировать. Гарантируется, что на текущий момент свободно хотя бы M домиков.

Если сейчас подается запрос второго типа, то вводится **Cancel Name**, где *Name* — название группы детей. Гарантируется, что в этот момент отменены заказы для всех групп, для которых они поступили раньше. Также гарантируется, что для этой группы заказ все еще не отменен.

Гарантируется, что сумма длин всех имен не превосходит 10^6 , а также сумма M по всем запросам не превышает 10^6 .

Формат выходных данных

Для каждого запроса первого типа выведите, какие домики будут забронированы для этой группы на момент запроса. Домики выводите в отсортированном порядке.

После всех запросов выведите все домики, которые остались не забронированы, тоже в отсортированном порядке.

Пример

стандартный ввод	стандартный вывод
5 3	1 2 3
Order dandelion 3	4
Order pear 1	1 2 3 5
Cancel dandelion	

Замечание

Обратите внимание на то, что решения, отправленные на руру могут работать быстрее, чем решения, отправленные на python.

Также при отправке на python/руру в этой задаче стоит использовать быстрый ввод. Для этого добавьте в начало кода следующие строчки:

```
import sys  
def input(): return sys.stdin.readline()
```

Задача С. Неправильная яблоня

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

Власти города Невозможного занимаются не только переукладыванием асфальта, но и посадкой новых деревьев. Было решено высадить яблоневую аллею. Чтобы всё было красиво, деревья в такой аллее должны располагаться в одну линию по убыванию высоты. Для аллеи заказали так много саженцев, что они не влезли в один грузовик. Поставщик направил три грузовика, в которых было N_1 , N_2 и N_3 деревьев. Для удобства транспортировки, саженцы в каждом из грузовиков располагались по возрастанию высоты.

Вскоре, все деревья были высажены в красивую аллею, но оказалось, что на складе саженцев что-то перепутали, и центральное дерево вовсе не яблоня, поэтому его решили заменить. Но чтобы аллея оставалась красивой, новое дерево должно быть не выше ближайшего левого и не ниже ближайшего правого. Укажите диапазон высоты саженца, который нужно заказать для замены центрального дерева.

Формат входных данных

В первой строке входных данных вводится число N_1 ($0 < N_1 < 4 \cdot 10^5$) — количество саженцев в первом грузовике.

На следующей строке через пробел вводится неубывающая последовательность a_1 ($0 < a_{1_i} < 10^8$) из N_1 чисел — высоты саженцев в первом грузовике.

На третьей и четвёртой строках аналогично вводятся число N_2 ($0 < N_2 < 4 \cdot 10^5$) и последовательность a_2 ($0 < a_{2_i} < 10^8$) из N_2 чисел.

На пятой и шестой строках аналогично вводятся число N_3 ($0 < N_3 < 4 \cdot 10^5$) и последовательность a_3 ($0 < a_{3_i} < 10^8$) из N_3 чисел.

Гарантируется, что $N_1 + N_2 + N_3$ — нечётное.

Формат выходных данных

Выведите через пробел два натуральных числа. Сначала минимальную возможную высоту нового саженца, затем — максимальную.

Пример

стандартный ввод	стандартный вывод
3	11 15
4 10 11	
6	
2 6 15 16 21 28	
4	
3 15 22 24	

Задача D. Множество с запросами

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

У вас изначально множество S , которое изначально пустое. Также есть q запросов трех типов:

1. Добавить какое-то число в S . Гарантируется, что такого числа еще нет в множестве.
2. Удалить какое-то число из S . Гарантируется, что такое число есть в множестве.
3. Рассмотрим функцию $f(A) = \max(A) - \frac{\text{sum}(A)}{2}$, где $\max(A)$ — максимальное число в A , а $\text{sum}(A)$ — сумма чисел в A . Необходимо найти и вывести максимальное значение функции $f(A)$ по всем подмножествам S .

Формат входных данных

В первой строке дано число q — количество запросов ($1 \leq q \leq 3 \cdot 10^5$). Затем следуют q строк — сами запросы.

Если запрос первого типа, то будет введено 1 val — это означает, что надо добавить число val ($1 \leq val \leq 10^9$). Гарантируется, что такого числа еще нет. Гарантируется, что val — это четное число.

Если запрос второго типа, то будет введено 2 val — это означает, что надо удалить число val ($1 \leq val \leq 10^9$). Гарантируется, что такое число сейчас есть в множестве.

Если запрос третьего типа, то будет введено 3 — после этого надо вывести максимальное значение $f(A)$ по всем подмножествам S . Гарантируется, что в этот момент S не пустое.

Формат выходных данных

На каждый запрос типа 3 выведите максимальное значение функции, а затем перевод строки.

Примеры

стандартный ввод	стандартный вывод
5 1 30 3 1 40 1 10 3	15 20
9 1 18 1 28 1 8 3 2 28 2 8 3 1 40 3	14 9 20

Задача Е. Максимизируй AND

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

Мальчик Петя недавно узнал в школе о новой для него операции `and` — операции побитового «И». Для того, чтобы понять, как устроена эта операция, рассмотрим пример. Например, мы хотим вычислить значение `14 and 10`. Для этого запишем числа 14 и 11 в двоичной системе счисления: $12_{10} = 1100_2$, а $9_{10} = 1001_2$. Теперь для того, чтобы вычислить значение `12 and 9`, нужно для каждой пары соответствующих битов вычислить их `and`: `1 and 1 = 1`, `1 and 0 = 0`, `0 and 0 = 0`, `0 and 1 = 0`. Таким образом, мы получаем ответ: $1000_2 = 8_{10}$.

Также Пете на день рождения, который был совсем недавно, подарили последовательность, состоящую из n целых чисел a_1, a_2, \dots, a_n . Пете хочется найти *подотрезок* этой последовательности с максимальной *and-суммой*.

Подотрезком последовательности называется набор чисел a_l, a_{l+1}, \dots, a_r для некоторых l и r , таких что $1 \leq l \leq r \leq n$.

and-суммой подотрезка a_l, a_{l+1}, \dots, a_r называется величина равная a_l `and` a_{l+1} `and` ... `and` a_r .

Формат входных данных

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

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

Формат выходных данных

Выведите одно целое число — величину максимальной *and-суммы* среди всех подотрезков последовательности a .

Примеры

стандартный ввод	стандартный вывод
1 5	5
2 1000 7	1000

Замечание

Последовательность из первого примера имеет только один подотрезок, который состоит ровно из одного элемента.

Последовательность из второго примера имеет три подотрезка: $[1000]$, $[7]$ и $[1000, 7]$. Значения *and-сумм* этих подотрезков равны 1000, 7 и 0, соответственно.

Задача F. День рождения

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

Завтра у Кагуи день рождения, поэтому Миюки решил придумать для нее *особенную* игру.

Миюки принесет с собой три ящика, которые будут заполнены конфетами: в первом будут лежать a конфет, во втором — b конфет, а в третьем — c конфет. В игре участвуют два человека. За один ход необходимо взять любое количество конфет, большее нуля, из одного ящика. Конечно же, невозможно забрать больше конфет, чем сейчас есть в ящике. Проигрывает тот, кто при очередном ходе не сможет забрать ни одной конфеты.

Так как Миюки старается приготовить *особенный* подарок, у игры тоже есть некая *особенность*: гарантируется, что изначально в каких-то двух ящиках количество конфет было одинаково.

Первый ход делает Кагуя. Определите, сможет ли она выиграть вне зависимости от ходов соперника.

Формат входных данных

Единственная строка содержит три целых числа a , b и c ($1 \leq a, b, c \leq 10^9$) — количество конфет в первом, втором и третьем ящиках, соответственно.

Формат выходных данных

Если Кагуя сможет победить, в первой строке выведите слово «Yes» (без кавычек).

Во второй строке через пробел выведите описание первого хода, который должна сделать Кагуя, чтобы выиграть. Выведите два числа i и k ($1 \leq i \leq 3$), где i — номер ящика, а k — количество конфет, которые должна взять Кагуя из ящика с номером i , чтобы обеспечить себе победу. Если существует несколько выигрышных первых ходов, выведите любой из них.

В противном случае в единственной строке выведите слово «No» (без кавычек).

Пример

стандартный ввод	стандартный вывод
3 3 7	YES 3 7

Задача G. Диагональный прямоугольник

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Дед Мороз подарил Васе на Новый год необычный прямоугольник. Вася сначала очень расстроился, но прямоугольник оказался действительно необычным. Оказалось, две его противоположные стороны можно сжимать и растягивать до любых размеров, так что он остается прямоугольником. Две другие стороны, длины которых равны a (известно, что a делится на 3 или на 5), всегда остаются фиксированными и изменить их размер не получится. Вася начал ставить эксперименты: он изменял длину той стороны, которую можно сжимать и растягивать. Оказалось, что эту сторону можно изменить так, чтобы ее длина стала целым числом и длина диагонали прямоугольника тоже стала целым числом. Как только Вася получает любой такой прямоугольник, он радуется и считает его площадь.

В задаче могут использоваться сведения из геометрии, а именно теорема Пифагора. По теореме Пифагора, если a , b и c — стороны прямоугольного треугольника, где катеты a и b примыкают к прямому углу, а c — наибольшая сторона (гипотенуза), то выполняется $a^2 + b^2 = c^2$

Формат входных данных

В единственной строке содержится число a ($3 \leq a \leq 10^8$) — длина фиксированной стороны прямоугольника. Выполняется условие, что a делится на 3 или на 5.

Формат выходных данных

В единственной строке выведите число S — площадь любого прямоугольника, удовлетворяющего условиям. (Так как площадь может быть большим числом, то на C++ рекомендуется использовать тип `long long`)

Примеры

стандартный ввод	стандартный вывод
3	12
5	60
9	108

Задача Н. Задача каменного века

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

Когда-то давно Миша и Миша решили придумать интересную задачу на очередной этап РОИ (редкая олимпиада по информатике). Один из них придумал прототип задачи, но другой своровал идею и предложил задачу на другой этап этой же олимпиады. С тех пор первый Миша ждал возможности предложить оригинальную идею на какую-либо другую олимпиаду... Ждал Миша до этого момента!

Вам дан массив a из n целых чисел. Также даны q запросов двух типов:

- Заменить i -й элемент массива на число x .
- Заменить каждый элемент массива на число x .

После выполнения каждого запроса вы должны вычислить сумму всех элементов в массиве.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 2 \cdot 10^5$) — количество элементов в массиве и количество запросов, соответственно.

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

Каждая из следующих q строк содержит описание очередного запроса. Описание запроса начинается с целого числа t ($t \in \{1, 2\}$), которое обозначает тип запроса:

- Если $t = 1$, далее следуют два целых числа i и x ($1 \leq i \leq n, 1 \leq x \leq 10^9$) — позиция элемента, который нужно изменить, и его новое значение.
- Если $t = 2$, далее следует целое число x ($1 \leq x \leq 10^9$) — число, на которое нужно заменить каждый элемент массива.

Формат выходных данных

Выведите q целых чисел, каждое в отдельной строке. В i -й строке нужно вывести сумму всех элементов массива после выполнения первых i запросов.

Пример

стандартный ввод	стандартный вывод
5 5	19
1 2 3 4 5	50
1 1 5	51
2 10	42
1 5 11	5
1 4 1	
2 1	

Замечание

Рассмотрим массив из примера и результат выполнения каждого запроса:

1. Изначально массив равен $[1, 2, 3, 4, 5]$.
2. После выполнения первого запроса массив равен $[5, 2, 3, 4, 5]$. Сумма всех элементов равна 19.
3. После выполнения второго запроса массив равен $[10, 10, 10, 10, 10]$. Сумма всех элементов равна 50.

4. После выполнения третьего запроса массив равен $[10, 10, 10, 10, 11]$. Сумма всех элементов равна 51.
5. После выполнения четвертого запроса массив равен $[10, 10, 10, 1, 11]$. Сумма всех элементов равна 42.
6. После выполнения пятого запроса массив равен $[1, 1, 1, 1, 1]$. Сумма всех элементов равна 5.

Задача I. Меховые подпоследовательности

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

Недавно Миша устал от бесконечной прокрастинации и захотел решить какую-нибудь задачу. К сожалению, он неправильно прочитал первую же найденную им задачу, а ошибку заметил уже после того, как отправил решение на проверку. К счастью, изначальная задача оказалась несложной, и Миша все же решил ее. А теперь вам предстоит решить его задачу!

Вам дан массив из n неотрицательных целых чисел. Ваша задача — посчитать количество подпоследовательностей i_1, i_2, \dots, i_k , таких что $1 \leq i_1 < \dots < i_k \leq n$, а также для каждого $1 \leq j \leq k$ выполнено неравенство $|a_{i_j} - M| \leq 1$, где $M = \text{mex} \{a_{i_1}, \dots, a_{i_k}\}$.

Напомним, что $\text{mex} \{a_1, \dots, a_n\}$ — это минимальное целое неотрицательное число, которое отсутствует в множестве чисел $\{a_1, \dots, a_n\}$. Например, $\text{mex} \{0, 1, 2\} = 3$, а $\text{mex} \{2, 3\} = 0$.

Формат входных данных

Первая строка содержит единственное целое число n ($1 \leq n \leq 10^5$) — количество элементов в массиве.

Вторая строка содержит n целых неотрицательных чисел a_1, \dots, a_n ($0 \leq a_i \leq 10^9$) — элементы массива a .

Формат выходных данных

Выведите одно целое число — количество искомых подпоследовательностей i_1, i_2, \dots, i_k .

Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Пример

стандартный ввод	стандартный вывод
4 0 2 3 2	4

Замечание

Рассмотрим пример из условия. В нем подходят следующие последовательности индексов:

1. [1]. Это множество $\{0\}$, его mex равен 1.
2. [1, 2]. Это множество $\{0, 2\}$, его mex равен 1.
3. [1, 4]. Это множество $\{0, 2\}$, его mex равен 1.
4. [1, 2, 4]. Это множество $\{0, 2\}$, его mex равен 1.

Задача J. AvtoBus

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

Наступила весна, и руководство автобусного парка AvtoBus дало распоряжение поменять у всех автобусов зимние шины на летние.

Вы владеете небольшой фирмой, занимающейся обслуживанием автобусов, и только что вам поступил долгожданный заказ на замену n шин. Вы знаете, что автобусный парк владеет автобусами двух типов: с двумя осями (у таких автобусов 4 колеса) и с тремя осями (у таких автобусов 6 колес).

Вы не знаете, сколькими автобусами какого типа владеет парк AvtoBus, поэтому вам стало интересно, сколько автобусов может быть в парке. Вам нужно определить, какое минимальное и какое максимальное количество автобусов может быть в парке, зная, что суммарное количество колес у всех автобусов равно n .

Формат входных данных

В единственной строке записано целое число n ($1 \leq n \leq 10^{18}$) — суммарное количество колес у всех автобусов.

Формат выходных данных

В единственной строке выведите два целых числа x и y ($1 \leq x \leq y$) — минимальное и максимальное возможное количество автобусов, которыми владеет парк.

В случае, если для данного n не существует подходящего количества автобусов, выведите в качестве ответа число -1 .

Примеры

стандартный ввод	стандартный вывод
4	1 1
7	-1
24	4 6
998244353998244352	166374058999707392 249561088499561088

Замечание

В первом примере известно, что суммарное количество колес у всех автобусов равно 4. Это значит, что в автобусном парке есть ровно один автобус с двумя осями.

Во втором примере не трудно показать, что не существует такого количества автобусов, чтобы их суммарное количество колес было равно 7.

В третьем примере суммарное количество колес у всех автобусов равно 24. Возможны следующие варианты:

- Четыре автобуса с тремя осями.
- Три автобуса с двумя осями и два автобуса с тремя осями.
- Шесть автобусов с двумя осями.

Таким образом, минимальное количество автобусов равно 4, а максимальное — 6.

Задача К. Счастливым порядком

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

Рустам занимается изучением натуральных чисел. В какой-то момент его заинтересовали *счастливые* числа. *Счастливыми* Рустам называет натуральные числа, десятичная запись которых состоит только из цифр 4 и 7.

Мальчик начал исследовать различные способы упорядочить эти числа. Он решил, что хочет упорядочить все счастливые числа *лексикографически*.

Так как счастливых чисел бесконечно много, Рустам просит вас помочь ему с исследованием и найти для него n -е счастливое число в лексикографическом порядке.

Напомним, что строка a *лексикографически меньше* строки b , если выполнено одно из двух условий:

1. Строка a является префиксом строки b (и при этом $a \neq b$).
2. Существует такое i ($1 \leq i \leq \min(|a|, |b|)$), что $a_i < b_i$, и для любого j ($1 \leq j < i$) $a_j = b_j$.

Число x лексикографически меньше числа y , если десятичная запись числа x лексикографически меньше десятичной записи числа y . Для определенности будем считать, что десятичная запись любого числа не содержит ведущих нулей.

Формат входных данных

В единственной строке содержится целое число n ($1 \leq n \leq 10^6$).

Формат выходных данных

Выведите n -е счастливое число в лексикографическом порядке.

Пример

	стандартный ввод	стандартный вывод
1		4