

Задача А. Самое частое слово

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

Дан текст. Выведите слово, которое в этом тексте встречается чаще всего.

Если таких слов несколько, выведите то, которое меньше в лексикографическом порядке.

Формат входных данных

Вводится текст.

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
apple orange banana banana orange	banana
oh you touch my tralala mmm my ding dingdingg	ding

Задача В. Степени двойки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Вам задано n чисел a_1, a_2, \dots, a_n . Найдите количество пар индексов i, j ($i < j$) таких, что $a_i + a_j$ является степенью двойки (то есть найдется такое целое число x , что $a_i + a_j = 2^x$).

Формат входных данных

В первой строке следует целое положительное число n ($1 \leq n \leq 10^5$) — количество чисел.

Во второй строке следует n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите количество пар индексов i, j ($i < j$) таких, что $a_i + a_j$ является степенью числа 2.

Примеры

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

Замечание

В первом примере в ответ входят следующие пары индексов: $(1, 4)$ и $(2, 4)$.

Во втором примере каждая пара индексов (i, j) (где $i < j$) входит в ответ.

Задача С. А и В и ошибки компиляции

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

А и В готовятся к олимпиадам по программированию.

В очень любит отлаживать свой код. Но перед тем, как запустить решение и начать отладку, код нужно сначала скомпилировать.

Изначально компилятор выдавал n ошибок компиляции, каждая из которых обозначается положительным целым числом. После некоторых усилий, В удалось исправить сначала одну, а потом еще одну ошибку.

Однако, несмотря на то, что В уверен, что исправил две ошибки, он не может понять, какие именно ошибки компиляции исчезли — компилятор языка, на котором пишет В, каждый раз выдает ошибки в новом порядке! В уверен, что в отличие от многих других языков программирования, ошибки компиляции для его языка программирования не зависят друг от друга, т. е. от исправления одной ошибки, множество остальных ошибок не изменяется.

А вы можете помочь В узнать, какие именно две ошибки он исправил?

Формат входных данных

Первая строка входных данных содержит целое число n ($3 \leq n \leq 10^5$) — изначальное количество ошибок компиляции.

Вторая строка содержит n целых чисел через пробел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — номера ошибок, выданных компилятором в первый раз.

Третья строка содержит $n - 1$ целых чисел через пробел b_1, b_2, \dots, b_{n-1} — номера ошибок, выданных при второй компиляции. Гарантируется, что последовательность в третьей строке содержит все числа второй строки, за исключением ровно одного.

Четвертая строка содержит $n - 2$ целых чисел через пробел c_1, c_2, \dots, c_{n-2} — номера ошибок, выданных при третьей компиляции. Гарантируется, что последовательность в четвертой строке содержит все числа третьей строки, за исключением ровно одного.

Формат выходных данных

Выведите два числа на отдельной строке: номера ошибок компиляции, исчезнувших после того как В внёс первое и второе исправление соответственно.

Примеры

стандартный ввод	стандартный вывод
5 1 5 8 123 7 123 7 5 1 5 1 7	8 123
6 1 4 3 3 5 7 3 7 5 4 3 4 3 7 5	1 3

Замечание

В первом тестовом примере в первый раз В исправляет ошибку с номером 8, во второй — с номером 123.

Во втором тестовом примере в первый раз В исправляет ошибку с номером 1, во второй — с номером 3. Обратите внимание, что если имеются несколько ошибок с одинаковым номером, за один раз В может исправить только одну из них.

Задача D. Симметрическая разность множеств

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

Реализуйте бинарную операцию над множествами — симметрическую разность. Ее результатом являются те элементы двух множеств, которые принадлежат ровно одному из множеств (но не принадлежат их пересечению).

Формат входных данных

В первой строке записано натуральное число — количество элементов первого множества.

Во второй строке через пробел перечислены эти элементы (натуральные числа).

В третьей строке указано количество элементов второго множества (натуральное).

В четвертой строке перечислены через пробел элементы второго множества (натуральные числа).

Формат выходных данных

Результат требуется вывести в том же формате: сначала количество элементов, полученных в результате операции, затем (если получено ненулевое количество элементов) вывести сами элементы в порядке возрастания.

Множества во входных данных могут быть неупорядочены и содержать равные элементы. Множество в выводе должно быть упорядочено и все элементы его должны быть различными.

Пример

стандартный ввод	стандартный вывод
3	1
1 2 3	3
2	
1 2	

Задача E. Множество

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

Реализуйте множество с использованием ассоциативного массива.

Формат входных данных

В первой строке содержится натуральное число n ($1 \leq n \leq 200000$) — количество запросов. В следующих n строках введены запросы трёх видов:

- Запрос вида '+ x ' означает добавить число x в множество, если его там нет;
- Запрос вида '- x ' означает удалить число x из множества, если оно там есть;
- Запрос вида '? x ' означает вопрос: есть ли x в множестве на момент выполнения запроса?

Значения всех чисел по модулю не превосходит 10^9 .

Формат выходных данных

На каждый запрос третьего типа в отдельной строке выведите 1, если число есть, и 0 в противном случае.

Пример

стандартный ввод	стандартный вывод
7	1
+ 7	0
? 7	0
- 7	0
? 7	
? 10	
- 10	
? 10	

Задача F. Словарь

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

У банка есть клиенты. Каждый клиент имеет ровно один счёт.

Напишите программу (обязательно с использованием ассоциативного массива), которая будет выполнять последовательность запросов таких двух видов:

1. Запрос вида 1 *name money*, где *name* — имя клиента, состоящее только из латинских букв, а *money* — сумма денег, которая добавляется к счёту текущего клиента. Сумма — целое число, по модулю не превышающее 10000.
2. Запрос вида 2 *name*. На каждый такой запрос программа должна ответить, какая сумма в данный момент есть на счёту у заданного клиента. Если такое имя клиента пока ни разу не упоминалось в запросах вида 1, выводите вместо числа слово 'ERROR'.

В начале работы программы у всех клиентов на счёту 0. Затем суммы на счёту у клиентов могут становиться как положительными, так и отрицательными.

Обратите внимание, что в ситуации, когда клиент снял суммарно столько же денег, сколько положил, сумма на счёте становится равной 0; но, раз его имя уже встречалось, нулевое значение не является основанием вывести 'ERROR'.

Формат входных данных

Первая строка стандартного входного потока — количество запросов N ($1 < N < 100000$). Далее следует N строк, в каждой из которых описан один из двух описанных выше видов запроса. Длина каждого имени не превосходит 20. Каждое имя состоит из латинских букв.

Формат выходных данных

На каждый запрос 2-го вида нужно вывести текущее значение на счёту у заданного клиента (или слово 'ERROR').

Пример

стандартный ввод	стандартный вывод
10	100
1 andrew 100	0
2 andrew	ERROR
1 andrew -100	-20000
2 andrew	-20000
2 zhenya	
1 anton -10000	
1 anton -10000	
2 anton	
2 anton	
1 alisa 10000	

Задача G. Игра в перерыве

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

Война войной, но и отдых тоже нужен. Поэтому Рей, Финн и Хан Соло после очередного тяжелого дня собрались вместе и решили поиграть в одну занимательную игру. Один из игроков выписывал на доску n чисел a_i , после этого с ними можно было производить следующие действия:

- Взять любое четное число из выписанных и вместо него написать на доску два числа, в два раза меньших выбранного;
- Два одинаковых выписанных числа заменить на одно, равное их сумме.

Выигрывает тот, кто сможет предъявить последовательность действий, которая максимизирует наибольшее число на доске. Вы стали случайным свидетелем этой игры, и вам стало интересно, чему же равно это наибольшее число.

Формат входных данных

В первой строке содержится число n ($1 \leq n \leq 100\,000$) — количество чисел, выписанных изначально на доске.

Во второй строке содержится n чисел a_i ($1 \leq a_i \leq 2 \cdot 10^9$) — числа, изначально выписанные на доску.

Формат выходных данных

В единственной строке выведите максимальное число, которое можно получить с помощью описанных выше операций.

Система оценки

Первая группа тестов состоит из тестов, для которых выполняются ограничения:

- $1 \leq n \leq 1000$;
- $1 \leq a_i \leq 2^{20}$ для всех i ;
- $a_i = 2^k$ для всех i (k является целым числом).

Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Вторая группа тестов состоит из тестов, для которых выполняются полные ограничения. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 60 баллов.

Примеры

стандартный ввод	стандартный вывод
4 1 3 3 6	12
6 2 2 2 2 4 4	16

Задача Н. Минимальный индекс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Пусть последовательность a_0, a_1, \dots задана таким образом:

$$\begin{cases} a_0 = (b * x_0 + y_0) \bmod M; \\ x_i = (c * a_{i-1} + x_{i-1}) \bmod M; \\ y_i = (d * a_{i-1} + y_{i-1}) \bmod M; \\ a_i = (b * x_i + y_i) \bmod M; \end{cases}$$

Здесь b, x_0, y_0, c, d, M — заданные константы. Выведите наименьший индекс этой последовательности, где нашёлся повторяющийся элемент, и расстояние между этими одинаковыми элементами, если он нашёлся среди первых 10^6 членов последовательности, и выведите 'GREATER' в противном случае.

Формат входных данных

Во входных данных в одной строке заданы шесть чисел — b, x_0, y_0, c, d, M ($0 \leq b, x_0, y_0, c, d \leq 10^9, 1 \leq M \leq 10^9 + 9$).

Формат выходных данных

Выведите два числа — индекс первого повторившегося элемента и расстояние до первого вхождения этого элемента в последовательности, если он нашёлся среди первых 10^6 членов последовательности, или 'GREATER' в противном случае.

Примеры

стандартный ввод	стандартный вывод
1 2 3 4 5 6	2 1
111 91 10 8 1 100001	9090 9090

Задача I. Встречалось ли число раньше

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

Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово *YES* (в отдельной строке), если это число ранее встречалось в последовательности или *NO*, если не встречалось.

Формат входных данных

Сначала задано число n — количество элементов в массиве ($1 \leq n \leq 100000$). Далее через пробел записаны n чисел — элементы массива. Массив состоит из целых чисел a_i ($-2^{31} \leq a_i \leq 2^{31} - 1$).

Формат выходных данных

Для каждого числа выведите слово *YES* (в отдельной строке), если это число ранее встречалось в последовательности или *NO*, если не встречалось.

Примеры

стандартный ввод	стандартный вывод
6 1 2 3 2 3 4	NO NO NO YES YES NO
10 1 1 1 1 1 1 1 1 1 1	NO YES YES YES YES YES YES YES YES
10 1 2 3 4 5 6 7 8 9 10	NO NO NO NO NO NO NO NO NO
10 10 10 4 2 5 5 10 4 1 10	NO YES NO NO YES YES YES NO YES

Задача J. Количество различных чисел

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

Вводится массив, состоящий из целых чисел. Определите, сколько в нем встречается различных чисел.

Примечание. Эту задачу на Питоне можно решить в одну строчку.

Формат входных данных

Сначала задано число n — количество элементов в массиве ($1 \leq n \leq 100000$). Далее через пробел записаны n чисел — элементы массива. Массив состоит из целых чисел a_i ($-2^{31} \leq a_i \leq 2^{31} - 1$).

Формат выходных данных

Необходимо вывести количество различных чисел в массиве.

Примеры

стандартный ввод	стандартный вывод
12 1 1 1 1 1 1 1 1 1 1 1 1	1
11 1 2 3 4 5 1 2 1 2 7 3	6
10 1 2 3 4 5 6 7 8 9 10	10
5 1 2 3 2 1	3