

Задача А. Вы все уже победители

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

Подготовка олимпиады — не только создание задач, но и набор серьёзных организационных вопросов.

Задачи олимпиады уже подготовлены, причём так, что самую сложную задачу способны решить только n конкретных школьников региона.

К сожалению, у каждого такого школьника может быть от 0 до k друзей, которые придут на олимпиаду и смогут списывать у него решения задач.

По рекомендациям министерства, среди участников олимпиады должен быть хотя бы один участник, не решивший самую сложную задачу. При каком минимальном числе участников это требование гарантированно будет выполнено?

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 10\,000$).

Во второй строке содержится целое число k ($0 \leq k \leq 10\,000$).

Формат выходных данных

Выведите единственное целое число — ответ на поставленную задачу.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 3 | 9 |
| 6 0 | 7 |

Задача В. Заработай себе металлолома

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

В одной замечательной игре «Рис» основной валютой является **металлолом**. Заработать его можно по-разному, но профессиональный игрок Вася придумал оригинальный способ.

После недавнего патча Вася заметил, что одна из мини-игр стала потенциально уязвимой с точки зрения лёгкого заработка.

Мини-игра заключалась в следующем: пусть у игрока изначально имеется **металлолом** в количестве k . В игре есть круглый барабан, разбитый на n равных секторов и имеющий единственную стрелочку. i -й сектор содержит целое положительное число x_i . Числа могут повторяться. Перед вращением барабана игрок делает ставку: сопоставляет некоторым x_i какое-то **целое положительное** количество **металлолома** b_j , причём сумма всех b_j не может превосходить k . Заметим, что ставка делается **на число**, то есть если у нас несколько секторов помечены одним и тем же числом, то ставка всё равно делается один раз.

После того, как ставка была сделана, вращается барабан. Число, на котором после вращения барабана остановится стрелочка, считается выигрышным. Игрок получает в качестве выигрыша количество **металлолома**, равное произведению выигрышного числа на соответствующее значение b_j . Ставки на остальные поля «сгорают». В частности, если игрок ничего не сопоставил выигрышному числу, то он теряет весь поставленный **металлолом**.

Помогите Васе определить, можно ли поставить весь свой **металлолом** в количестве k штук так, чтобы после игры **гарантированно** получить **больше металлолома**, чем у него было изначально.

Формат входных данных

В первой строке через пробел записаны два целых числа n и k — количество секций барабана и изначальное кол-во **металлолома** у Васи, соответственно ($1 \leq n \leq 10^5, 1 \leq k \leq 10^9$).

Во второй строке через пробел записаны n целых чисел x_i — числа, записанные на секторах барабана ($1 \leq x_i \leq 10^9$).

Формат выходных данных

Выведите в качестве ответа на поставленный вопрос одну строку: «YES», если Вася может сделать ставку так, чтобы гарантированно увеличить количество **металлолома**, либо «NO» в противном случае.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------|-------------------|
| 5 12 2 2 4 4 4 | YES |
| 5 12 2 2 4 4 5 | NO |
| 3 100 2 3 4 | NO |

Замечание

В первом тесте Вася может сопоставить числу 2 7 **металлолома**, а числу 4 5 **металлолома**. При выигрышном числе 2 Вася получит 14 **металлолома**, а при выигрышном числе 4 20 **металлолома**.

Задача С. Обледенение

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1.5 секунд |
| Ограничение по памяти: | 256 мегабайт |

Представим лобовое стекло автомобиля в виде сетки из n строк и m столбцов. На пересечении строки i и столбца j стекло покрыто льдом толщиной $c_{i,j}$.

Толщина 0 означает, что клетка не покрыта льдом и даёт беспрепятственный обзор, иначе клетка не даёт обзора вообще. Чтобы начать поездку, необходимо, чтобы хотя бы k клеток давали обзор.

У стекла имеется обогрев, который за одну секунду уменьшает толщину льда на 1 в каждой клетке. При этом, если образовался кусок льда, состоящий из нескольких клеток, который не касается рамки стекла, то он падает, тем самым исчезая со стекла. Заметим, что этот кусок пропадает моментально и никак не влияет на другие льдинки. Две клетки принадлежат одному куску льда, если они имеют общую сторону и их толщина больше 0. Кусок льда считается касающимся рамки стекла, если в нём есть клетка из крайней строки и/или из крайнего столбца.

Выясните сколько секунд нужно подождать с момента начала обогрева, чтобы можно было начинать поездку.

Формат входных данных

Первая строка содержит целые числа n , m и k — размеры стекла и необходимая видимость, соответственно ($2 \leq n, m \leq 450$; $1 \leq k \leq n \times m$).

В следующих n строках содержатся по m чисел $c_{i,j}$, разделённых пробелом ($1 \leq c_{i,j} \leq 10^9$).

Формат выходных данных

Выведите количество секунд, через которое стекло будет обладать необходимой видимостью.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 5 5 9 9 9 9 9 9 9 2 1 3 9 9 1 9 2 9 9 4 5 1 9 9 9 9 9 9 | 5 |
| 3 3 6 9 8 7 4 5 6 3 2 1 | 6 |

Задача D. Дата сайнс

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

Назовём последовательность чисел x_i *пилообразной*, если $x_1 > x_2 < x_3 > x_4 < \dots$ или $x_1 < x_2 > x_3 < x_4 > \dots$ (заметим, что последовательность из одного числа и последовательность из двух различных чисел являются пилообразными).

Назовём *хаотичностью* последовательности x_i наибольшее число k такое, что можно удалить некоторые числа из x и оставшиеся будут образовывать пилообразную последовательность размера k .

Михаилу, начинающему дата-саентисту, необходимо обработать данные, представленные в виде массива n **различных** чисел a_i . Есть два типа запросов к данным:

- «! i x» — установить a_i равным x (гарантируется, что после этого все a_i будут различны);
- «? l r» — вычислить хаотичность отрезка $a[l \dots r]$, то есть последовательности a_l, a_{l+1}, \dots, a_r ;

По загадочным обстоятельствам Михаил больше не будет программировать алгоритмы, поэтому ему нужна ваша помощь в обработке данных.

Формат входных данных

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

Вторая строка содержит n целых чисел a_i ($1 \leq a_i \leq 10^9$).

Третья строка содержит целое число q — количество запросов ($1 \leq q \leq 300\,000$).

В каждой из следующих q строк содержится описание запроса в формате, описанном выше. Для запросов изменения выполняется $1 \leq i \leq n$ и $1 \leq x \leq 10^9$. Для запросов на отрезке выполняется $1 \leq l \leq r \leq n$.

Гарантируется, что в любой момент a состоит из попарно различных чисел.

Формат выходных данных

Для каждого запроса второго типа (то есть начинающегося с «?») в порядке их поступления на вход выведите в новой строке одно целое число — ответ на этот запрос.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 | 6 |
| 1 6 2 5 3 4 | 4 |
| 7 | 2 |
| ? 1 6 | 4 |
| ? 2 5 | |
| ! 1 9 | |
| ! 4 1 | |
| ? 1 4 | |
| ! 1 5 | |
| ? 1 6 | |

Задача Е. Вирусные воспоминания

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

Учёные наконец-то научились считывать воспоминания из мозга, теперь самые яркие воспоминания за последние n дней можно представить в виде массива целых чисел так, что одинаковые числа соответствуют одним и тем же воспоминаниям и наоборот, разные числа соответствуют разным воспоминаниям.

Среди последовательности некоторых воспоминаний назовём воспоминание x *затмевающим*, если x встречается в этой последовательности чаще всех остальных воспоминаний вместе взятых. Например, в последовательности $[1, 2, 2, 1, 2]$ воспоминание 2 будет затмевающим, а в последовательности $[3, 1, 2, 3]$ ни одно таковым не является.

Также учёные доказали, что мозг способен менять воспоминания. Если существует отрезок подряд идущих воспоминаний такой, что x является затмевающим среди них на данный момент, то за одну операцию мозг может заменить каждое воспоминание с этого отрезка на x .

Назовём воспоминание x *вирусным*, если за некоторое, возможно ноль, количество операций мозг может сделать все воспоминания равными x .

По заданной последовательности воспоминаний сообщите обо всех вирусных воспоминаниях.

Формат входных данных

Первая строка содержит целое число n — количество воспоминаний ($1 \leq n \leq 200\,000$).

Вторая строка содержит n целых чисел t_i — типы воспоминаний ($1 \leq t_i \leq n$).

Формат выходных данных

В первую строку выведите m — количество вирусных воспоминаний.

Во второй строке выведите m целых различных чисел в **любом порядке** — типы таких воспоминаний.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 1 2 2 3 2 3 | 2 2 3 |
| 4 3 1 2 3 | 0 |

Задача F. Докраска забора

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

Поликарпыч приехал на свою дачу и первым делом заметил свой старый недокрашенный забор. Некоторые доски забора уже были окрашены и Поликарпыч решил докрасить его так, чтобы все окрашенные доски, как старые, так и новые, образовывали непрерывный отрезок досок в заборе.

Поликарпыч может купить в магазине кисть ширины W , что позволит ему за один вертикальный мазок красить сразу W подряд идущих досок, где W — любое натуральное число.

К сожалению, из-за особенности местной экологии, никакую доску нельзя красить повторно, в том числе и свежеекрашенную.

Поликарпыч хочет купить одну кисть и с её помощью докрасить забор. Определите минимальное количество мазков, за которое это можно сделать.

Формат входных данных

Первая строка содержит целое число n — количество уже окрашенных досок ($1 \leq n \leq 300\,000$).

Вторая строка содержит n целых чисел x_i — номера досок ($1 \leq x_i \leq 10^{18}$). Гарантируется, что номера досок не повторяются.

Формат выходных данных

Выведите минимальное количество мазков для докраски забора.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 3 5 8 | 3 |
| 4 1 4 5 8 | 2 |

Замечание

В первом примере $W = 1$, красить нужно доски с номерами 4, 6 и 7.

Во втором примере при $W = 2$ за первый мазок можно покрасить доски с номерами 2 и 3, а за второй — 6 и 7.

Задача G. Скатерть

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

Стол является прямоугольником со сторонами A и B . Его нужно накрыть квадратной скатертью так, чтобы каждая точка стола была закрыта и стороны скатерти были расположены к сторонам стола под 45 градусов.

Вычислите минимальную площадь такой скатерти.

Формат входных данных

В единственной строке содержатся два целых числа A и B ($1 \leq A, B \leq 10\,000$).

Формат выходных данных

Выведите минимальную площадь скатерти, удовлетворяющей требованиям.

Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не превосходит 10^{-9} .

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 2 | 8 |
| 3 6 | 40.5 |
| 1234 5678 | 23887872 |

Задача Н. История одного анекдота

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

Хорошо известно, что анекдоты меняются из уст в уста и проследить за всеми изменениями довольно сложно.

Пользуясь методами современной анекдотологии, любой анекдот можно представить в виде слова, состоящего из строчных букв латинского алфавита, а изменение анекдота может быть одного из двух типов:

- 1) «push_x» — добавить букву «x» в конец слова;
- 2) «pop» — удалить последнюю букву (этот тип применим только для непустых слов).

Анекдотолог Иван собрал историю одного анекдота — все версии анекдота, полученные корректной последовательностью изменений начиная с пустого слова, но есть нюанс. К сожалению, версии даны в произвольном порядке, а для публикации в научном журнале нужно восстановить последовательность изменений.

Помогите Ивану — найдите последовательность изменений такую, что применив её к изначально пустому слову, набор всех промежуточных версий совпадёт с имеющейся историей не учитывая порядка.

Формат входных данных

Первая строка содержит целое число n — количество версий ($1 \leq n \leq 3000$).

В следующих n строках содержится по одному слову, состоящему из строчных букв латинского алфавита и записанному в кавычках.

Суммарная длина слов не превосходит 10^6 .

Гарантируется, что слова получены корректной последовательностью изменений, описанных в условии задачи, начиная с пустого слова.

Формат выходных данных

Выведите список из $n - 1$ изменений в том порядке, в котором их нужно выполнять. Каждое изменение должно быть описано в отдельной строке.

Список слов, полученных этим списком действий, должен совпадать с данным списком слов, не учитывая порядка.

Примеры

| стандартный ввод | стандартный вывод |
|--|--|
| 5 "a" "b" "" "" "" | push_a pop push_b pop |
| 6 "" "" "a" "b" "ab" "a" | push_b pop push_a push_b pop |

Задача I. Кружок

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

На занятиях кружка олимпиадного программирования «ШФМ Лаго» доступно n компьютеров, пронумерованных от 1 до n и расположенных по кругу.

Для компьютера с номером i следующим будет являться компьютер с номером $i + 1$, а предыдущим — с номером $i - 1$. При этом, для компьютера n следующим является 1-й, а для компьютера 1 предыдущим является n -й.

В начале учебного года на кружок записалось m учеников. Известно, что i -й ученик в первый день займёт компьютер с номером s_i , а в каждый последующий день будет пересаживаться. Если $d_i = 1$, то он будет занимать следующий компьютер, а если $d_i = -1$, то предыдущий.

Преподаватель кружка опасается, что если два ученика в один из дней захотят занять один и тот же компьютер, то из-за возможных последствий кружок закроют навсегда. Поэтому, придётся принять в кружок только таких учеников, у которых **независимо от количества занятий** в кружке не возникнет конфликта ни в какой из дней занятий.

Выясните, какое наибольшее количество учеников смогут заниматься в кружке.

Формат входных данных

Первая строка содержит целые числа n и m — количество компьютеров и количество учеников, соответственно ($2 \leq n \leq 10^9$; $1 \leq m \leq 200\,000$).

В следующих m строках содержатся числа s_i и d_i — номер компьютера в первый день и направление, соответственно ($1 \leq s_i \leq n$; $d_i = 1$ или $d_i = -1$).

Формат выходных данных

Выведите единственное целое число — наибольшее количество учеников, которых можно принять в кружок.

Примеры

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

Задача J. JuryGPT

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

Жюри олимпиад решило снять с себя ответственность за обработку вопросов по задачам во время олимпиады и предоставить это искусственному интеллекту. С его помощью, участники могли бы получать точные ответы без томительного ожидания.

К сожалению, интегрировать цельный GPT модуль в тестирующую систему не представляется возможным. Поэтому, именно Вы должны реализовать легковесный аналог — «JuryGPT»!

Вашей программе будет дано условие задачи и вопрос участника. Требуется определить, что следует ответить участнику: «Read problem statement» («Читайте условие задачи») или «No comments» («Без комментариев»).

Подойдите к реализации ответственно, ведь именно Ваша разработка может появиться на следующем соревновании!

Формат входных данных

В первой строке содержится условие задачи. Во второй — вопрос участника.

Формат выходных данных

Выведите ответ на вопрос участника: «Read problem statement» или «No comments».

Примеры

| стандартный ввод | стандартный вывод |
|---|------------------------|
| Даны два натуральных целых числа. Выведите их сумму. Здравствуйте, у меня появился вопрос: могут ли числа быть отрицательными? | Read problem statement |
| Найдите наибольший общий делитель двух целых чисел. А что такое наибольший общий делитель? | No comments |
| Мама подарила Пете на день рождения массив чисел... Нет ли ошибки в тестах? Я не понимаю где у меня может быть ошибка... | No comments |