

Задача А. Арсенал

Имя входного файла: `arsenal.in`
Имя выходного файла: `arsenal.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во время очередной тренировки к играм Китнисс решила потренироваться в стрельбе из лука.

В тренировочном зале есть n стрел разной длины, которые находятся в стойках рядом друг с другом вдоль стены. Китнисс хочет выстрелить все стрелы, при этом после каждого выстрела ей придется брать новую стрелу из стойки, а взятые стрелы никогда не возвращаются обратно. Китнисс считает, что стрелу удобно взять, если рядом с ней с обеих сторон стоят стрелы, размеры которых меньше. Отсутствие стрелы с одной из сторон равносильно присутствию стрелы с длиной ноль.

Чтобы процесс тренировки прошел наиболее эффективно Китнисс хочет взять как можно меньше неудобных стрел. Но так как, ей нужно тренироваться, она просит вас ей помочь подобрать порядок выбора стрел из стоек, чтобы количество неудобно взятых стрел было наименьшим возможным.

Формат входного файла

В первой строке входного файла дано одно натуральное число n ($1 \leq n \leq 10^5$) — количество стоек со стрелами.

Во второй строке дано n чисел a_i ($1 \leq a_i \leq 1000$) — высота стрел.

Формат выходного файла

В первой строке выходного файла выведите наименьшее количество стрел, которые придется неудобно взять.

Во второй строке через пробел выведите n чисел — номера стрел в порядке, в котором Китнисс будет их брать. Если способов взять стрелы несколько, выведите любой.

Пример

<code>arsenal.in</code>	<code>arsenal.out</code>
4 1 2 3 4	0 1 3 2 4
4 1 2 1 1	1 1 2 3 4
6 1 1 2 2 3 3	3 1 2 4 3 5 6

Задача В. Лук Пита

Имя входного файла: `bow.in`
Имя выходного файла: `bow.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пит хочет похвастаться своим луком перед Китнисс. Конечно же, самое главное в луке — это дальность стрельбы.

Чтобы показать Китнисс, насколько далеко стреляет его лук, Пит встал в точку с координатой 0 и выстрелил по прямой. Стрела упала в точку X прямой, по которой стрелял Пит. Пит знает число X , и предлагает выяснить его Китнисс. Для этого он дал Китнисс n подсказок вида: $x_i d_i$, означающих, что точка x_i находится от точки X на расстоянии не более чем d_i .

Теперь по этой информации Китнисс хочет выяснить, какое наибольшее X удовлетворяет всем подсказкам Пита.

Формат входного файла

В первой строке содержатся одно натуральное число n ($1 \leq n \leq 10^5$).

В следующих n строках содержатся по два целых числа $x_i d_i$ ($0 \leq x_i \leq 10^9$, $0 \leq d_i \leq 10^9$).

Формат выходного файла

В единственной строке выведите наибольшее X , удовлетворяющее всем подсказкам Пита, либо -1 , если не существует X , удовлетворяющего всем подсказкам Пита.

Пример

<code>bow.in</code>	<code>bow.out</code>
2 1 2 3 2	3
3 2 2 6 2 8 1	-1
5 100 97 115 104 97 115 111 115 107 97	197

Задача С. Цвета

Имя входного файла: `colors.in`
Имя выходного файла: `colors.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Мало кто знает, что Цинна с детства увлекается изобразительным искусством. Раньше он часами проводил время, стоя перед холстом с палитрой в руках. Вот и сейчас, увидя давно забытую кисть, он с радостью вспоминает правило, которое позволяет получить огромное количество удивительных цветов: из цветов с номерами a и b можно получить цвет с номером $\lfloor \frac{a+b}{2} \rfloor$. Цинну стало интересно, сколько различных цветов можно получить из имеющихся. Помогите ему разобраться с этим!

Формат входного файла

В первой строке входного файла находится число n ($1 \leq n \leq 10^5$) — количество цветов, имеющихся у Цинны.

Во второй строке входного файла находятся n чисел a_i ($1 \leq a_i \leq 10^9$) — номера цветов.

Цвета могут повторяться.

Можно считать, что у Цинны имеется неограниченное количество заданных цветов.

Обратите внимание, что операцию можно применять и над получившимися в результате другой операции цветами.

Формат выходного файла

Выведите максимальное количество различных цветов, которые может получить Цинна.

Примеры

<code>colors.in</code>	<code>colors.out</code>
6 5 7 10 12 8 4	9

Задача D. Засада

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

Пробираясь к Капитолию, отряд Китнисс попал в засаду — на них выскочило 1000 вооруженных солдат Капитолия и приказало сдаться. Однако Китнисс и ее солдаты не собираются это делать, поэтому они тут же выхватили оружие и направили его на солдат Капитолия. Китнисс понимает, что каждый человек из ее отряда направил свое оружие на случайного человека — на того, на кого получилось. Также она уверена, что если начнется стрельба, все выстрелы ее солдат попадут в цель, то есть каждый из них убьет солдата противника, на которого направил оружие.

Теперь надо оценить, стоит ли начинать стрельбу, а именно, надо понять, сколько в этом случае солдат противника будет убито. За одну секунду Китнисс может спросить у двух своих солдат, на разных ли солдат они направили свое оружие. Помогите ей найти количество солдат противника, которое будет убито в случае начала перестрелки не более чем за 40 000 секунд.

Протокол взаимодействия с программой жюри:

В самом начале программа жюри сообщает вашей программе число n ($1 \leq n \leq 1000$) — количество солдат в отряде Китнисс.

Дальше во время взаимодействия вашей программы с программой жюри несколько раз повторяются следующие действия:

- ваша программа сообщает программе жюри число *type*
- Если *type* = 1, вы также должны сообщить два числа i, j ($1 \leq i, j \leq n$) — это означает, что вы хотите получить от солдат с номерами i и j информацию, на одного ли солдата Капитолия они направили оружие.
- В случае *type* = 2 вам необходимо вывести ответ на задачу и сразу же завершить программу.
- программа жюри сообщает вашей программе:
 - «-1», если номер солдата Капитолия, на которого направил оружие солдат номер i , меньше, чем номер солдата, на которого направил оружие солдат номер j
 - «0», если эти номера равны
 - «1», если номер i -го солдата больше номера j -го
- в случае, если в вашем запросе *type* = 1, описанные действия начинают повторяться сначала

Пример

stdin	stdout
1 1 3	3
1 2 3	-1
1 1 2	-1
2 2	0

Комментарий

В тесте из условия номера солдат, на которых были направлено оружие, были равны 1, 1 и 2 соответственно.

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача Е. Слежка от президента

Имя входного файла:	<code>notepad.in</code>
Имя выходного файла:	<code>notepad.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Президент Сноу любит шпионить за своими друзьями и врагами. Чтобы узнать о чем переписывается Плутарх, президент приказал поставить на его компьютер программу, которая будет сообщать какие кнопки нажимает Плутарх.

На компьютере Плутарха открыто n текстовых окон, в каждом отображается текстовый файл. На момент начала работы шпионской программы все окна пустые. Также известно, что в окне отображаются последние k символов текстового документа. Есть несколько кнопок, которые он нажимает:

- Латинские строчные буквы. В конец текущего текстового файла добавляется символ, соответствующей нажатой кнопки.
- **Backspace**. Из конца текущего текстового документа удаляется один символ. Если документ пустой, ничего не происходит.
- **Copy**. Та часть текстового файла, которая видна на экране сохраняется в буффер обмена.
- **Paste**. В конец текущего текстового файла добавляется то, что лежит в буффере обмена. Изначально буффер пуст.
- **Next**. Открывает следующее окно. Порядок обхода окон всегда одинаковый. После последнего окна кнопка открывает первое.

Полученные данные нужно обработать. Наверняка все самое ценное будет написано на экране после нажатия всех кнопок. Президент не пожалеет для вас ни одну розу, если вы решите для него эту задачу.

Формат входного файла

В первой строке входного файла дано три натуральных числа n , m , k ($1 \leq n \leq 10; 1 \leq m, k \leq 1000$) — количество окон на компьютере Плутарха, количество нажатий клавиш и ограничение на количество отображаемых в окне символов.

В следующих n строках даны описания нажатых клавиш — строчные латинские буквы, либо название клавиши из описанных выше.

Формат выходного файла

В единственной строке выходного файла выведите информацию, которая будет отображена на экране компьютера после нажатия всех клавиш. Если текущий текстовый файл пустой выведите `Empty`.

Пример

notepad.in	notepad.out
2 7 10 Next a b Copy Next Paste Paste	abab
10 7 10 a b Copy Backspace Paste Paste Backspace	aaba
1 7 3 a Copy Paste Copy Paste Copy Paste	aaa
3 5 10 a b Next c Next	Empty

Задача F. Построение

Имя входного файла: `parade.in`
Имя выходного файла: `parade.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Перед началом игры традиционно проводят построение всех участников. Все участники становятся в одну шеренгу и объявляется старт. В построении должны участвовать все n участников. Каждый из которых должен одеть одеяние некоторого цвета.

Для того чтобы построение было наиболее зрелищным, шеренга должна удовлетворять следующему требованию: задаются m отрезков $[l_i \dots r_i]$ в шеренге, после построения на каждом таком отрезке цвета одеяний всех участников должны быть различны.

От вас требуется найти такие цвета одеяний, которые удовлетворяют этому требованию, причем минимизировав количество различных использованных цветов.

Формат входного файла

В первой строке содержатся два натуральных числа n, m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$).

В следующих m строках содержатся отрезки $[l_i \dots r_i]$ ($1 \leq l_i \leq r_i \leq n$).

Формат выходного файла

В первой строке выведите натуральное число k — минимальное количество различных использованных цветов.

В следующей строке выведите n чисел a_i ($1 \leq a_i \leq k$) — цвет одеяния i -го участника. Если существует несколько ответов — выведите любой.

Пример

<code>parade.in</code>	<code>parade.out</code>
3 1 2 3	2 2 2 1
6 3 2 3 4 6 3 6	4 3 3 2 3 1 4
dabcba 5 3 4 5 2 2 3 3	2 1 1 1 1 2

Задача G. Телепорты

Имя входного файла: `teleports.in`
Имя выходного файла: `teleports.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сейчас строится новая арена для очередных голодных игр. Одним из ключевых элементов на ней будет система телепортов. Телепорты будут располагаться на окружности длины l . Так же, телепорты не будут включены все время: иногда некоторые будут выключаться или включаться.

Для определения увлекательности игры важным является максимальное время, которое может понадобиться на то, чтобы добраться из одной точки на окружности до другой, передвигаясь исключительно по окружности. Считается, что в среднем трибут бежит со скоростью v . Для передвижения он может либо бежать по окружности в любом направлении, либо, если он находится в точке с работающим телепортом, воспользоваться им. При этом, он мгновенно перемещается, по своему желанию, в любой, работающий на данный момент, телепорт.

Вы знаете, как будет изменяться состояние телепортов. Для каждого состояния выведите максимальное время, которое может понадобиться на то, чтобы добраться от одной точки окружности до другой, если трибут использует оптимальный маршрут.

Позиции телепортов задаются расстоянием по окружности по часовой стрелке от определенной фиксированной точки.

Формат входного файла

В первой строке находятся четыре целых числа n, m, l и v ($1 \leq n, m \leq 10^5$, $3 \leq l \leq 10^9$, $1 \leq v \leq 1000$, $n \leq l$) — количество изначально включенных телепортов, количество изменений состояний телепортов, длина окружности и скорость трибута.

В следующей строке находятся n различных целых чисел x_i ($0 \leq x_i < l$) — позиции телепортов, включенных в начале.

В следующих m строках находится описание изменений состояний телепортов.

Если строка начинается с символа «+», в этой строке находится описание включения телепорта. Далее в этой строке находится целое число y_i ($0 \leq y_i < l$) — позиция включаемого телепорта.

Если строка начинается с символа «-», в этой строке находится описание выключения телепорта. Далее в этой строке находится целое число y_i ($0 \leq y_i < l$) — позиция выключаемого телепорта.

Гарантируется, что при включении телепорта, на этой позиции телепорт не включен, и что при выключении телепорта, на этой позиции телепорт включен.

Формат выходного файла

В $(m + 1)$ строке выведите ответы для каждого из состояний.

В первой строке выведите ответ для начального состояния.

В $(i + 1)$ -й строке выведите ответ для состояния после i изменений ($1 \leq i \leq m$).

Ответ будет считаться правильным, если он выведен с абсолютной или относительной погрешностью не более 10^{-6} .

Пример

teleports.in	teleports.out
3 4 100 3	13.3333333333
5 25 75	16.6666666667
- 5	14.1666666667
+ 90	12.1666666667
+ 37	14.6666666667
- 75	

Задача Н. Обучение войск

Имя входного файла: `warriors.in`
Имя выходного файла: `warriors.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Перед наступлением всегда важно быть уверенным в своей победе, а для этого все солдаты должны быть хорошо подготовлены ко всему. Сегодня на утреннем осмотре Китнисс с Питом оценили боеспособность своего войска. Для каждого солдата они выписали на лист бумаги число от 1 до 3 — его подготовленность (1 — полностью готов к бою, 3 — ничего не умеет). Но на самом деле не всегда для победы нужна обученность именно всего войска. Китнисс придумала тактику, при которой они точно победят, если хотя бы половина их бойцов полностью готова к бою. Посмотрев на листок с выписанными характеристиками солдат, предводительница задалась вопросом: какое минимальное количество бойцов надо еще полностью подготовить к бою, чтобы, благодаря ее тактике, победа им была обеспечена? Помогите ей найти это количество, ведь времени на раздумья у нее совсем нет.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 100\,000$) — количество солдат.

Во второй строке входного файла содержится информации на листке бумаги — n чисел a_i ($1 \leq a_i \leq 3$).

Формат выходного файла

В единственной строке выходного файла выведите минимальное количество солдат, которых надо полностью подготовить к бою для безоговорочной победы.

Примеры

<code>warriors.in</code>	<code>warriors.out</code>
3 1 2 3	1
3 1 1 2	0

Задача I. Выбор оружия

Имя входного файла: `weapons.in`
Имя выходного файла: `weapons.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Победа уже близко, и Китнисс планирует последнее наступление. Нужно много людей и, конечно же, оружия. Оружие бывает двух типов: автоматическое (автоматы, пистолеты, пулеметы) и метательное (ножи, лук, арбалет). Каждый человек по-своему умеет обращаться с обоими типами оружия. Так, для каждого солдата Китнисс узнала два числа — его эффективность обращения с автоматическим оружием и метательным соответственно. Теперь ей надо каждому человеку выдать ровно один тип оружия, причем понятно, что сделать она это хочет таким образом, чтобы суммарная эффективность обращения с оружием всех людей была максимальна. Однако, и отправлять в бой одних лучников или одних автоматчиков — не лучшая идея. Поэтому Китнисс также решила, что количество людей, которым она даст метательное оружие и людей, которым она даст автоматическое оружие, будет отличаться не более чем на m .

Формат входного файла

В первой строке входного файла дано два числа n, m ($1 \leq n \leq 100\,000, 1 \leq m \leq n$) — количество солдат, готовых идти в бой, и максимальная разница количества людей, которые получают автоматическое оружие, и людей, которые получают метательное оружие соответственно.

В следующих n строках заданы умения людей обращаться с оружием. $i + 1$ -я строка входного файла содержит два числа a_i, b_i ($1 \leq a_i, b_i \leq 10\,000$) — эффективность обращения i -го человека с автоматическим и метательным оружием соответственно.

Формат выходного файла

В первой строке выходного файла выведите суммарную эффективность обращения с оружием все солдат. Во второй строке выведите n чисел, i -е из которых равно 0 или 1 — 0 означает, что солдату номер i дали автоматическое оружие и 1, если метательное.

Примеры

<code>weapons.in</code>	<code>weapons.out</code>
2 1 1 2 1 2	3 1 0
3 1 1 2 1 3 2 1	7 1 1 0

Комментарий

В первом тестовом примере оба солдата одинаково владеют метательным и автоматическим оружием. Однако, дать им обоим автоматическое оружие нельзя — количество людей с автоматическим оружием и количество людей с метательным оружием должно отличаться не более чем на 1.

Во втором тестовом примере опять же нельзя дать всем трем солдатам автоматическое оружие, поэтому оптимальный вариант — дать первым двум из них автоматическое, а третьему — метательное, тогда суммарное умение будет равно $2 + 3 + 2 = 7$.