

Задача А. Убить босса

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

Лунтик играет в новую видеоигру. Он практически прошел все уровни, ему остался последний, на котором необходимо убить босса. Босс имеет h единиц здоровья, а Лунтик за один удар отнимает ровно a единиц здоровья. Известно, что босс никак не может пополнить запас здоровья и погибает, когда у него остаётся ноль единиц здоровья или меньше.

Сколько ударов потребуется нанести Лунтику, чтобы победить босса?

Формат входных данных

В первой строке дано одно натуральное число h ($1 \leq h \leq 10^4$) — исходное число единиц здоровья босса. Во второй строке дано натуральное число a ($1 \leq a \leq 10^4$) — урон от одного удара Лунтика.

Формат выходных данных

Выведите одно число — сколько ударов потребуется нанести Лунтику, чтобы победить босса?

Примеры

стандартный ввод	стандартный вывод
10 4	3
1 10000	1
10000 1	10000

Замечание

В первом примере Лунтику требуется три удара. После первого удара здоровье босса равно 6. После второго удара здоровье босса равно 2. После третьего удара здоровье босса равно -2 .

Задача В. Таблица умножения

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

Выучив таблицу умножения, Лунтик теперь знает, как перемножить два целых числа от 1 до 9 (включительно).

По данному числу n определите, может ли n быть представлено, как произведение двух целых чисел от 1 до 9.

Формат входных данных

Вам дано одно натуральное число n ($1 \leq n \leq 100$).

Формат выходных данных

Если возможно представить число n в виде произведения двух целых чисел от 1 до 9, выведите «Yes», если не может, выведите «No» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
10	Yes
50	No
81	Yes

Задача С. Светофор

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

Давайте посмотрим на самый распространенный автомобильный светофор, состоящий из трех секций (сверху вниз): красной, желтой и зеленой лампочки. Каждой секции может быть сопоставлено два цвета: соответствующий ей цвет во включенном состоянии и черный цвет в выключенном состоянии. Когда светофор работает правильно, существует пять возможных состояний. Они подчиняются следующему алгоритму:

1. Горит только зеленый свет;
2. Мигает зеленый свет;
3. Зеленый гаснет, горит желтый;
4. Желтый гаснет, загорается красный;
5. Горит красный и загорается желтый;
6. Желтый и красный гаснут и все повторяется с пункта 1.

Помните, что светофор может работать в режиме, когда есть только желтый мигающий сигнал.

Напишите программу управления светофором, которая в соответствии с текущей индикацией сигналов светофора определяет следующее состояние, в которое он должен перейти, или устанавливает, что светофор поврежден.

Формат входных данных

Первая строка ввода описывает текущее состояние верхней секции светофора, вторая строка — средней секции, третья строка — нижней секции. Статус каждой секции определяется его цветом: «b» (черный), «r» (красный), «y» (желтый) и «g» (зеленый). Если цвет мигает, то его название пишется заглавными буквами, иначе — строчными.

Формат выходных данных

В трех строках стандартного вывода программа должна отобразить описание следующего состояния светофора следующим образом: если участок черный, напишите «black», если он красный — «red», зеленый — «green», желтый — «yellow». При необходимости используйте заглавные буквы, чтобы обозначить мигающий сигнал.

Если светофор не работает, выведите «error». Все надписи следует выводить без кавычек.

Примеры

стандартный ввод	стандартный вывод
b b g	black black GREEN
b Y b	black YELLOW black
r Y g	error

Задача D. Все на своих местах

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

У Лунтика есть n одинаковых карточек. Он написал целое число на каждой карточке. Числа от 1 до n написаны таким образом, что каждое число встречается ровно один раз. Он расположил карты в ряд, поместив карту, на которой было написано число 1, на первое место, карту, на которой было написано число 2, на второе место и так далее. Лунтик отправился на луну, а затем пришел Кузя, который поменял некоторые карточки местами.

Когда Лунтик вернулся с луны, он заинтересовался, какое наименьшее и какое наибольшее число написаны на тех карточках, которые лежат не на своём месте. Помогите ему, напишите программу, которая находит эти числа.

Формат входных данных

В первой строке дано целое число n ($3 \leq n \leq 1000$) — количество карточек. Во второй строке через пробел даны n целых чисел, разделенных пробелами — числа на карточках после того, как их передвинул Кузя.

Формат выходных данных

В одной строке стандартного вывода ваша программа должна вывести два целых числа, разделенных пробелом — наименьшее и наибольшее число, которое интересует Лунтика.

Когда нет чисел, которые интересуют Лунтика, то есть все карточки лежат на своих местах, ваша программа должна вывести два нуля, разделенные одним пробелом.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	0 0
5 1 5 3 4 2	2 5
7 6 5 4 3 2 1 7	1 6

Задача Е. Автобусы

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

Лунтик вырос и начал работать программистом в новой компании. Однажды, когда он ехал на работу, ему пришлось долго стоять на остановке в ожидании автобуса.

Он знает, что новый человек прибывает на остановку каждые y минут, а новый автобус каждые x минут. Автобус может забрать с этой остановки не более n человек. Также известно, что количество автобусов, которые пройдут через остановку в этот день, равно p , а количество людей, которые сядут в автобусы, равно t (к сожалению, Лунтик придет на остановку последним, т.е. перед ним будет $t - 1$ человек). Он хочет знать, сколько минут ему придется ждать с момента прибытия на автобусную остановку до момента, когда он сядет в автобус. Люди садятся в автобус в порядке прибытия на остановку (по очереди).

Если человек приходит на остановку, и в этот же момент приходит автобус, в который он может сесть, он садится в него.

Формат входных данных

В первой строке дано одно натуральное число p — количество автобусов. Во второй строке дано одно натуральное число t — количество людей. В третьей строке вводится натуральное число x — интервал движения автобусов. В четвертой строке содержится натуральное число y — период прибытия пассажиров на остановку.

Мы предполагаем, что первый автобус прибывает в минуту x , а первый человек прибывает на остановку в минуту y . В пятой строке вводится натуральное число n — максимальное число человек, которое может сесть в автобус на этой остановке. Все числа во вводе не превосходят 10^9 .

Формат выходных данных

В единственной строке программа должна вывести количество минут, которое Лунтик будет ждать, прежде чем сесть в автобус. Если Лунтик не сможет сесть ни в один автобус, программа должна вывести «No» без кавычек.

Примеры

стандартный ввод	стандартный вывод
2 5 7 2 3	4
2 5 5 2 3	0
2 5 4 2 3	No

Замечание

В первом примере на второй минуте придет первый человек. На четвертой минуте и на шестой приходят еще два человека, теперь их на остановке трое. На седьмую минуту прибывает автобус,

который может забрать всех троих пассажиров на остановке. На восьмой минуте приходит новый человек (четвертый по счету). На десятой минуте приходит Лунтик (последний). На четырнадцатой минуте подъезжает новый автобус, который забирает двух пассажиров. Лунтик пришел на остановку на десятой минуте и садится в автобус на четырнадцатой, таким образом он ждал четыре минуты.

Числа в задаче могут не влезать в 32-битный тип данных, используйте 64-битные типы данных для целых чисел (`long long` в C++, `int64` в Pascal, `long` в C# и Java). В Python это не требуется.

Задача F. Дорога со светофорами

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

Лунтик едет на своем грузовике по длинной прямой дороге со множеством светофоров. Для каждого светофора он знает, как долго будут гореть красный и зеленый свет (цикл повторяется бесконечно).

Когда Лунтик начинает свое движение в момент времени 0, все светофоры только что загорелись красным. Лунтик движется с постоянной скоростью — одна единица расстояния в секунду. Когда светофор красный, он останавливается и ждет, пока он не станет зеленым. Когда перед Лунтиком загорается зеленый, он немедленно едет дальше. Если Лунтик подъехал к светофору, на котором только что загорелся зеленый, он проезжает без остановки, а если только что загорелся красный, то он немедленно останавливается.

Напишите программу, которая определяет, сколько времени потребуется Лунтику, чтобы добраться до конца дороги. Начало дороги находится на нулевом расстоянии, а конец дороги — на расстоянии x .

Формат входных данных

Первая строка содержит два целых числа n и x ($1 \leq n \leq 100; 1 \leq x \leq 1000$) — количество светофоров на дороге и длину дороги.

Каждая из следующих n строк содержит три целых числа d, r и g ($1 \leq d \leq x, 1 \leq r, g \leq 100$), описывающих светофор: расстояние от начала дороги до светофора, количество секунд, которое горит красный и зеленый свет соответственно.

Светофоры даны в порядке возрастания d . Никакие два светофора не находятся в одной точке.

Формат выходных данных

В единственной строке выведите натуральное число — время (в секундах), за которое Лунтик должен достичь конца дороги.

Примеры

стандартный ввод	стандартный вывод
2 10 3 5 5 5 2 2	12
4 30 5 1 1 7 13 5 14 4 4 15 3 10	36
2 3 1 1 1 2 1 1	4

Замечание

В первом примере Лунтик будет ждать две секунды на первом светофоре. Затем он дойдет до второго светофора, который в момент его прибытия был зеленым, и Лунтик сможет немедленно проехать.

В третьем примере Лунтик приедет к первому светофору в тот момент, когда тот загорается зеленым, таким образом он проезжает его без остановки. Когда он доезжает до второго светофора, тот загорается красным, поэтому Лунтик ждет одну секунду.

Задача G. Пасьянс

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

В пасьянсе используется n карт. На каждой карточке написано натуральное число от 1 до 100000. Возможно, есть несколько карточек, на которых написано одно и то же число. Три карты образуют триплекс, если на двух из них написаны равные числа, а на третьей написано число, равное сумме чисел на двух других картах. Например, карточки, на которых написаны числа 2, 4 и 2, образуют триплекс независимо от порядка, в котором они встречаются.

Изначально все карты размещаются так, чтобы числа, написанные на них, не были видны. В пасьянс играют карты одна за другой. Все карты, которые уже перевернуты, лежат перед игроком на столе, и их номера видны. Если при переворачивании очередной карты возникает триплекс с любыми из уже перевернутых карт, то участвующие в триплексе карты немедленно удаляются. Если более двух из уже перевернутых карт можно триплексировать с текущей, не имеет значения, какие две из них будут удалены. Например, если перед игроком уже лежат карты с номерами 4, 2, 4, 4 и очередная карта имеет номер 2, то за триплекс можно взять любую из карт, на которых написано число 4. Удаленные карты не участвуют в дальнейшем розыгрыше пасьянса. Пасьянс заканчивается после того, как перевернуты все возможные карты.

Напишите программу, которая определяет количество выпавших триплексов, количество карт, оставшихся перед игроком после завершения пасьянса, и наименьшее и наибольшее число, оставшиеся лежать на столе.

Формат входных данных

В первой строке дано натуральное число n ($3 \leq n \leq 10^6$) — количество карт, которые участвуют в пасьянсе. Во второй строке даны n натуральных чисел — достоинства карт в порядке их появления на столе. Достоинства карт не превосходят 10^5 .

Формат выходных данных

В первой строке выведите два целых числа, разделенных пробелом — количество триплексов и количество оставшихся перед игроком после пасьянса карт.

Во второй строке выведите два числа, разделенные пробелом — наименьшее и наибольшее из чисел на оставшихся на столе карточках. Если карт не осталось, выведите «NONE» без кавычек.

Примеры

стандартный ввод	стандартный вывод
11 8 4 2 6 2 1 4 9 6 6 4	2 5 1 9
9 3 6 6 7 3 12 14 7 6	3 0 NONE

Замечание

Во втором примере первая открытая карта - 3. Перед игроком лежит карта 3.

Вторая открытая карта 6, перед игроком лежат карты 3, 6. В них нет триплекса.

Третья открытая карта 6, перед игроком лежат карты 3, 6, 6. В них нет триплекса.

Четвертая открытая карта 7, перед игроком лежат карты 3, 6, 6, 7. В них нет триплекса.

Пятая открытая карта 3, перед игроком лежат карты 3, 6, 6, 7, 3. Карты 3, 6, 3 образуют триплекс (первый триплекс). Поскольку есть две карты с цифрой 6, берется любая из них. Карты 3, 3 и 6 удаляются, а перед игроком остаются 6, 7.

Шестая открытая карта 12, перед игроком лежат карты 6, 7, 12. В них нет триплекса.

Седьмая открытая карта 14, перед игроком лежат карты 6, 7, 12, 14. В них нет триплекса.

Восьмая открытая карта 7, перед игроком лежат карты 6, 7, 12, 14, 7. Карты 7, 14, 7 образуют триплекс (второй триплекс). Они удаляются, а перед игроком остаются 6, 12.

Девятая открытая карта 6, перед игроком лежат карты 6, 12, 6. Три карты образуют триплекс (третий триплекс) и удаляются. Перед игроком не осталось карт.

Задача Н. Четные цифры

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

Дано натуральное число n . Найдите наибольшее целое число, меньшее n , состоящее только из четных цифр.

Формат входных данных

В единственной строке дано натуральное число n ($1 \leq n \leq 10^{100}$).

Формат выходных данных

Выведите наибольшее целое число, меньшее n , состоящее только из четных цифр.

Пример

стандартный ввод	стандартный вывод
260	248

Задача I. Дальнобойщик

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

Лунтик устроился на работу дальнобойщиком. На контрольно-пропускных пунктах между странами формируются большие очереди грузовиков. Чтобы незаметно скоротать время ожидания, он придумывает различные логико-математические игры и задачи, которые затем пытается решить самостоятельно. Последняя игра, которую он придумал, связана с регистрационными номерами грузовиков, выстроенными в длинные очереди.

Лунтик помнит регистрационные номера всех n грузовиков, которые видит. Затем он пытается найти такое целое число x большее 1, что при делении на него все регистрационные номера грузовиков дают одинаковый остаток. Задача Лунтика — найти как можно больше таких чисел x . Помогите ему.

Формат входных данных

В первой строке вводится натуральное число n ($2 \leq n \leq 100$) — количество регистрационных номеров, запомненных Лунтиком. Каждая из следующих n строк содержит натуральное число — номер грузовика. Все числа разные и не превосходят 10^9 .

Формат выходных данных

В одной строке выведите все найденные числа, разделенные одним пробелом и расположенные в порядке возрастания (от меньшего к большему). Если таких чисел нет, выведите ноль.

Примеры

стандартный ввод	стандартный вывод
3 6 34 38	2 4
5 5 17 23 14 83	3
2 3 4	0

Замечание

В первом примере все числа дают остаток 0 при делении на 2, и остаток 2 при делении на 4.

Задача J. Сортировка обменами

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

Дана последовательность из n различных целых чисел $a_1, a_2, \dots, a_n$ со значениями в диапазоне от 0 до $n - 1$. Лунтик может выполнить операцию обмена, в которой он выбирает два числа и меняет их местами. Напишите программу, которая определяет, какое наименьшее количество обменов потребуется совершить Лунтику, чтобы расположить числа в порядке возрастания.

Формат входных данных

В первой строке дано одно натуральное число n ($3 \leq n \leq 10^5$) — длина последовательности. Вторая строка содержит n различных чисел в диапазоне от 0 до $n - 1$, разделенные пробелами.

Формат выходных данных

Выведите одно целое число, равное минимальному количеству обменов, необходимых для сортировки последовательности.

Пример

стандартный ввод	стандартный вывод
5 2 3 0 1 4	2

Замечание

Пояснение к примеру: меняем местами 2 и 0. Получаем последовательность 0, 3, 2, 1, 4. В получившейся последовательности меняем местами 3 и 1. Получаем последовательность 0, 1, 2, 3, 4. Больше никаких обменов не требуется, потому что у нас есть возрастающая последовательность.

Задача К. Мажоранта множества

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

Даны две последовательности из n натуральных чисел: $a_1, a_2, \dots, a_n$ и $b_1, b_2, \dots, b_n$. Мы говорим, что в первой последовательности есть мажоранта, если в ней есть элемент, который равен половине или более чем половине элементов этой последовательности. Не всегда в последовательности есть мажоранта.

Ваша задача состоит в том, чтобы получить мажоранту в первой последовательности, используя обмены элементов со второй последовательностью. За один обмен вы можете выбрать индекс некоторый индекс i , и поменять элементы a_i и b_i местами. Напишите программу, которая находит минимальное количество обменов, которое необходимо совершить, чтобы в первой последовательности появилась мажоранта.

Формат входных данных

В первой строке дано число T ($1 \leq T \leq 10$) — количество тестов, ответы на которые вам нужно найти. Каждый тест состоит из трёх строк, в первой дано натуральное число n ($1 \leq n \leq 2 \cdot 10^5$), в двух других даны по n натуральных чисел, не превосходящих 40 000, элементы последовательностей a и b .

Формат выходных данных

Для каждого теста выведите одно число на новой строке — минимальное необходимое число обменов. Если мажоранту никак нельзя получить, выведите -1 .

Пример

стандартный ввод	стандартный вывод
3	2
6	-1
3 2 1 4 9 10	0
5 5 3 5 3 1	
4	
1 2 3 4	
5 6 7 8	
5	
1 1 1 1 1	
2 2 2 2 2	

Замечание

Пояснение к первому тесту примера: меняем местами элементы в парах (a_3, b_3) и (a_5, b_5) . Таким образом, после двух обменов первая последовательность становится 3, 2, 3, 4, 3, 10, а значение 3 встречается в половине элементов последовательности.

Задача L. Изготовление моделей

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

Лунтик и Кузя увлеклись робототехникой и сумели создать специализированных роботов для изготовления моделей автомобилей. Лунтик создает роботов, которые собирают модели, а Кузя конструирует роботов, которые их раскрашивают и упаковывают. Начали производство в гараже Лунтика.

Помещение маленькое, и герои сначала устанавливают в гараже роботов, которые собирают модели. После того, как они собрали все модели, герои моментально убирают собирающих модели и ставят роботов, которые раскрашивают и упаковывают модели. У Лунтика и Кузи есть n различных роботов, собирающих модели. Роботы пронумерованы от 1 до n . Каждый из этих роботов сначала собирает первую модельку за a_i секунд, а каждую последующую модельку собирает уже за b_i секунд.

Как только все модельки собраны, Лунтик и Кузя моментально запускают m различных роботов, раскрашивающих и упаковывающих модели. Эти роботы пронумерованы от 1 до m . Каждый из этих роботов сначала раскрашивает и упаковывает модельку первую модельку за c_i секунд, а каждую последующую модельку раскрашивает и упаковывает уже за d_i секунд.

Луне очень нравится, как работают роботы второго типа, но она никогда не успевает приехать, чтобы за ними понаблюдать. Помогите ей, написав программу для расчета времени, когда второй тип роботов заработает, если вы знаете общее время сборки, покраски и упаковки, но не знаете количество сделанных моделей.

Формат входных данных

Первая строка содержит целое число t ($1 \leq t \leq 10^9$) — общее время, необходимое для сборки, покраски и упаковки моделей. Гарантируется, что герои используют роботов максимально эффективно, чтобы как можно быстрее сделать все модельки, они даже могут не включать некоторых роботов.

В следующей строке записано целое число n ($1 \leq n \leq 100$) — количество роботов первого типа. В каждой из следующих n строк записано по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq 10^9$) — время в секундах, которое требуется i -у роботу первого типа, чтобы подготовиться к работе и собрать первую модельку и время в секундах, которое необходимо i -у роботу первого типа для сборки каждой последующей модельки.

В следующей строке записано целое число m ($1 \leq m \leq 100$) — количество роботов второго типа. В каждой из следующих m строк записано по два целых числа c_i и d_i ($1 \leq c_i, d_i \leq 10^9$) — время в секундах, которое требуется i -у роботу второго типа, чтобы подготовиться к работе, затем раскрасить и упаковать первую модельку и время в секундах, которое необходимо i -у роботу второго типа, чтобы раскрасить и упаковать каждую последующую модельку.

Формат выходных данных

В одной строке стандартного вывода выведите целое число — количество секунд от старта роботов первого типа до старта роботов второго типа.

Примеры

стандартный ввод	стандартный вывод
12 1 3 1 1 5 1	5
20 2 3 2 1 3 3 3 1 4 1 5 1	13

Замечание

В первом примере герои изготовили 3 модельки, во втором примере 11 моделек.

Задача М. Цепочка простого числа

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

Положительное целое число называется простым, если оно больше 1 и делится только на 1 и само себя. Цепочкой простого числа p называется набор последовательных простых чисел, сумма которых равна p . Когда у простого числа нет цепочки описанного типа, мы считаем, что цепочка состоит из самого числа p . Например, 2, 3, 5, 7 — это цепочка для простого числа 17, потому что $17 = 2 + 3 + 5 + 7$. Число 7 представляет собой цепочку для самого себя, потому что простое число 7 не может быть представлено как сумма последовательных простых чисел меньших 7.

Напишите программу, которая находит простое число p на заданном отрезке $[a, b]$ (то есть p такое, что $a \leq p \leq b$), которое имеет самую длинную цепочку.

Формат входных данных

В единственной строке даны два натуральных числа a и b ($1 \leq a \leq b \leq 10^6$), которые определяют концы отрезка, на котором мы ищем простое число с описанным свойством.

Формат выходных данных

В первой строке ваша программа должна вывести найденное простое число и количество чисел в его цепочке.

Во второй строке ваша программа должна вывести числа из цепочки в возрастающем порядке. Если с искомым свойством имеется более одного простого числа, ваша программа должна вывести меньшее из них.

Если в данном числовом диапазоне нет простых чисел, ваша программа должна вывести только сообщение «no prime» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
10 20	17 4 2 3 5 7
24 28	no prime
42 52	43 1 43

Задача N. Массив дедушки Шера

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

Однажды, придя в гости к дедушке Шеру, Лунтик обнаружил на столе аккуратно разложенный массив натуральных чисел $a_1, \dots, a_n$. Лунтик — большой любитель простых чисел. Натуральное число называется **простым**, если оно имеет ровно два различных натуральных делителя: единицу и самого себя. Другими словами, число x является простым, если оно больше 1 и при этом делится без остатка только на 1 и на x .

Дедушка Шер решил дать Лунтику следующую задачу: выделить максимальное число непересекающихся пар индексов, таких что сумма чисел в массиве на соответствующих индексах есть простое число. Например, пусть у дедушки Шера есть массив 2, 8, 9, 3, 5, тогда Лунтик может составить две пары индексов: (1, 4) и (2, 3). Пары индексов (1, 3) и (2, 5) также подходят в данном случае.

Вскоре Лунтик понял, что это слишком сложная для него задача, поэтому он попросил вас помочь ему.

Формат входных данных

В первой строке вводится одно целое число n ($1 \leq n \leq 400$) — размер массива дедушки Шера.

Во второй строке вводятся n целых чисел $a_1, a_2, \dots, a_n$ ($2 \leq a_i \leq 10^6$), где a_i — i -е число в массиве дедушки Шера.

Формат выходных данных

Выведите одно целое число — максимальное количество пар.

Примеры

стандартный ввод	стандартный вывод
5 2 8 9 3 5	2
3 2 2 3	1