

## Задача А. Сбалансированные массивы

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Миша играет с массивом целых чисел. За один ход он может сделать с массивом следующие операции:

- Добавить 1 на каком-то суффиксе массива.
- Добавить 1 на каком-то префиксе массива.

Например, пусть Мишин массив — это  $(1, 2, 4)$ . Тогда за одну операцию он может получить один из следующих массивов:  $(2, 2, 4)$ ,  $(2, 3, 4)$ ,  $(2, 3, 5)$ ,  $(1, 2, 5)$ ,  $(1, 3, 5)$ .

Массив длины  $n$  называется *сбалансированным*, если Миша может получить этот массив из массива из  $n$  нулей, применив к нему какую-то последовательность операций. Например, массив  $(1, 2, 1)$  сбалансированный, а  $(1, 3, 1)$  — нет. Сколько существует сбалансированных массивов длины  $n$  с элементами не более  $m$ ? Ответ может быть большим, поэтому выведите его по модулю простого числа 998 244 353.

### Формат входных данных

Единственная строка входа содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 500\,000$ ).

### Формат выходных данных

Выведите одно число: количество сбалансированных массивов длины  $n$ , все элементы которых не больше  $m$ . Выведите его по модулю 998 244 353.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2	9

## Задача В. Корзины и шары

Имя входного файла: *стандартный ввод*  
 Имя выходного файла: *стандартный вывод*  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 512 мегабайт

У вас есть шарики, покрашенные в  $n$  различных цветов. Для каждого цвета  $i$  от 1 до  $n$  есть ровно  $x_i$  шариков этого цвета. Вы играете в игру, состоящую из действий. За одно действие вы можете взять  $k$  шариков попарно различных цветов и выкинуть их. Какое наибольшее количество действий вы можете совершить?

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $k$ : число цветов и количество шариков, выкидываемых при каждом действии ( $1 \leq k \leq n \leq 2 \cdot 10^5$ ). Во второй строке содержатся  $n$  чисел  $x_i$ , разделённых пробелами — количество шариков  $i$ -го цвета ( $1 \leq x_i \leq 10^9$ ).

### Формат выходных данных

Выведите строку, в которой будет одно целое число — наибольшее возможное количество действий.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 3 5 8 9 4	8
10 5 1 2 3 4 5 6 239 239 239 239	21

## Задача С. Игра в карты

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 15 секунд  
Ограничение по памяти: 512 мегабайт

Никита показывает фокусы с карточками. На каждой карточке написано целое число от  $-2$  до  $2$  включительно.

У Никиты есть шляпа фокусника с карточками, а также секретное число, которое изначально равно  $n$ . Никита повторяет следующую операцию  $m$  раз: он берёт случайную карточку из шляпы, добавляет число на карточке к своему секретному числу, а затем кладёт карточку обратно в шляпу.

Если секретное число стало отрицательным, Никита немедленно проигрывает игру. Он выигрывает, если не проиграл после  $m$  операций. Какова вероятность того, что он выигрывает?

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $0 \leq n, m \leq 100\,000$ ).

Вторая строка содержит пять целых чисел:  $x_{-2}$ ,  $x_{-1}$ ,  $x_0$ ,  $x_1$  и  $x_2$  ( $0 \leq x_i \leq 10^8$ ). Здесь  $x_i$  — это количество карточек, на которых написано  $i$ . В шляпе есть по крайней мере одна карточка.

### Формат выходных данных

Выведите одно целое число: вероятность по модулю простого числа 998 244 353. Другими словами, искомая вероятность — это рациональное число  $\frac{p}{q}$ . Вы должны вывести  $p \cdot q^{-1}$  по модулю 998 244 353.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1 1 1 1 1 1	399297742

### Замечание

В первом тесте искомая вероятность равна  $\frac{4}{5}$ .

## Задача D. Сказочные шахматы

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

У Алисы и Боба есть набор, состоящий из обычных и сказочных шахматных фигур. Всего в наборе 6 типов фигур (слон, ладья, ферзь, архиепископ, канцлер и магараджа). В наборе ровно две фигуры каждого типа. Ладья может перемещаться на любое количество клеток по горизонтали или вертикали. Слон может перемещаться на любое количество клеток по диагонали. Ферзь объединяет силу ладьи и слона и может перемещаться на любое количество клеток по горизонтали, вертикали или диагонали. Конь перемещается на любую из ближайших клеток, которые не находятся на той же горизонтали, вертикали или диагонали. Таким образом, ход образует форму буквы «Г»: две клетки по вертикали и одна клетка по горизонтали, или две клетки по горизонтали и одна клетка по вертикали. Архиепископ ходит как слон и конь, канцлер как ладья и конь, а магараджа как ферзь и конь.

Алиса и Боб решили сыграть в игру на шахматной доске 8 на 8. В начале игры фигуры перемещаются и выставляются в ряд, определяя порядок. Первую фигуру в заданном порядке на доску выставляет Алиса, вторую Боб, третью Алиса, и так далее. Фигуру нужно выставить на свободное поле так, чтобы она не била и не была под боем выставленных ранее фигур. Игрок, который не может сделать ход, проигрывает.

Ваша задача — написать программу, вычисляющую, кто победит при оптимальной игре обоих противников.

### Формат входных данных

Первая строка содержит 12 заглавных английских букв, задающих порядок выставления фигур на доску. «В» соответствует слону, «R» — ладье, «Q» — ферзю, «A» — архиепископу, «C» — канцлеру, «M» — магарадже. В заданном наборе ровно две фигуры каждого из шести типов.

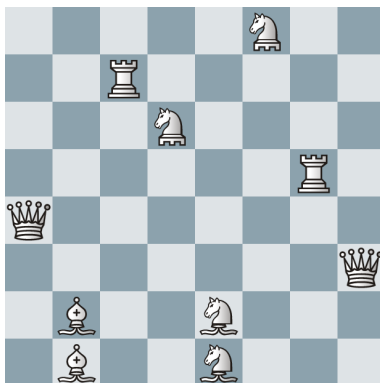
### Формат выходных данных

Выведите «Alice», если побеждает Алиса. Выведите «Bob», если побеждает Боб.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
BBAARRCCQMMM	Bob
BAMBAMQQRCCR	Alice

### Замечание



Возможная конечная позиция для расстановки из первого примера, если соперники играли не оптимально. Первые десять фигур (все, кроме магараджа) выставлены на доску. На d6 и f8 стоят канцлеры. На e1 и e2 стоят архиепископы. Одиннадцатую фигуру (магараджу) поставить некуда.

## Задача Е. Выборы

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

В Городе Правды начались выборы. На пост президента претендуют три кандидата, пронумерованные от 1 до 3. За них будут голосовать  $n$  избирателей. У каждого избирателя есть свой собственный список предпочтений, заданный перестановкой чисел 1, 2, 3. Например, перестановка (2, 3, 1) означает, что этот избиратель больше всего предпочитает кандидата 2, затем он предпочитает кандидата 3, а кандидат 1 является для него наименее предпочтительным. Списки предпочтений известны всем.

Избиратели будут голосовать (то есть выбирать одного кандидата) один за другим в установленном порядке, от первого избирателя до последнего избирателя. Более того, после того, как человек проголосует, он сразу же честно расскажет, как он проголосовал.

Президентом станет кандидат, которого выбрало большинство избирателей. Если несколько кандидатов набрали наибольшее количество голосов, победителем становится кандидат с наименьшим номером (например, если кандидаты 1 и 2 оба набрали наибольшее количество голосов, то побеждает кандидат 1).

Кто победит, если каждый избиратель проголосует оптимально?

### Формат входных данных

Первая строка содержит одно целое число  $n$ : количество избирателей ( $1 \leq n \leq 10\,000$ ).

Затем следуют  $n$  строк.  $i$ -я из этих строк содержит перестановку чисел 1, 2 и 3: предпочтения  $i$ -го избирателя.

### Формат выходных данных

Выведите одно целое число от 1 до 3: номер кандидата, который победит на выборах.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 2 1 1 2 3 2 1 3	2

## Задача F. Ровно три соседа

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Рассмотрим бесконечное во все стороны клетчатое поле. Каждая клетка покрашена либо в чёрный цвет, либо в белый. У каждой чёрной клетки **ровно три** соседние по стороне клетки — также чёрные.

Будем рассматривать периодические раскраски. А именно, сначала раскрасим какой-то прямоугольник из клеток. Затем разобьём поле на такие прямоугольники, стыкуя их по сторонам. Все прямоугольники раскрасим одинаково.

Приведите пример раскраски, в которой доля чёрных клеток равна заданному рациональному числу  $p/q$ , или выясните, что это невозможно.

### Формат входных данных

В первой строке заданы два целых числа  $p$  и  $q$  — числитель и знаменатель требуемой доли чёрных клеток ( $0 \leq p \leq q \leq 10$ ; числа  $p$  и  $q$  взаимно просты).

### Формат выходных данных

Если раскраска возможна, в первой строке выведите два целых числа  $h$  и  $w$ : высоту и ширину прямоугольника ( $1 \leq h, w \leq 1000$ ). Далее выведите раскраску прямоугольника из  $h$  строк, по  $w$  символов в каждой. Символ «.» (точка) обозначает белую клетку, символ «#» (решётка) — чёрную. Отношение количества чёрных клеток в прямоугольнике к общему количеству клеток в нём должно быть равно  $p/q$ . Если возможных раскрасок несколько, выведите любую из них.

Если раскраска невозможна, в первой строке выведите «-1 -1».

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>иллюстрация</i>
2 3	4 6 .####. ##..## ##..## .####.	
1 1	-1 -1	

## Задача G. Озеро

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Одним знойным летним днём Татьяна с ребятами пешком отправилась на озеро. Дорога была столь утомительна, что все обрадовались, когда за ними после купания внезапно прилетела на вертолёт их подруга Ольга.

В вертолёте доступны два ряда с пассажирскими сиденьями. Для уменьшения количества перелётов Оля разрешила взрослому к себе на колени сажать ребёнка на всех пассажирских сиденьях, кроме центрального заднего. А по центру заднего ряда сможет сесть либо один взрослый, либо один ребёнок.

Ниже представлена схема, на которой указано, на каких местах разрешено сидеть взрослому с ребёнком (максимально 2 пассажира на одном кресле), а где может располагаться максимум один человек: взрослый или ребёнок. А также отмечено место пилота — Ольги.

О.		max 2	
max 2	max 1	max 2	

Какое минимальное количество перелётов должна совершить Ольга, чтобы всех вернуть домой? Перемещение в одну сторону считается за отдельный перелёт.

### Формат входных данных

В первой строке через пробел записаны целые числа  $n$  и  $m$  — количество взрослых и детей, которые отправились на озеро пешком ( $1 \leq n, m \leq 10^6$ ).

### Формат выходных данных

Выведите минимальное количество перелётов для возвращения всех домой.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1	1
1 4	1
5 5	3

## Задача Н. Запрещённое множество

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Задано множество десятичных цифр. Найдите минимальное простое число, обладающее следующим свойством: в десятичной записи этого числа **нет** цифр из данного множества.

Например, если задано множество  $\{0, 6, 3, 9\}$ , то простое число 71 удовлетворяет требованию задачи (кроме, быть может, минимальности), а простое число 101 — нет (есть цифра 0, которая присутствует во множестве).

### Формат входных данных

Первая строка содержит одно целое число  $n$  — количество цифр во множестве ( $1 \leq n \leq 10$ ). Каждая из последующих  $n$  строк содержит одно целое число  $d_i$  ( $0 \leq d_i \leq 9$ ) — очередной элемент множества. Гарантируется, что все  $d_i$  попарно различны.

### Формат выходных данных

Если не существует простых чисел, в десятичной записи которых отсутствуют цифры из заданного множества, выведите  $-1$ . Иначе выведите наименьшее такое число.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
7 0 1 2 4 6 8 9	3
9 0 1 2 3 5 6 7 8 9	-1



## Задача I. Разноцветные циклы

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 8 секунд  
Ограничение по памяти: 1024 мегабайта

Дан связный неориентированный граф  $G$  без петель и кратных рёбер с  $n$  вершинами и  $m$  рёбрами. Каждое ребро имеет свой собственный цвет: красный, жёлтый или синий. Существует ли в графе простой цикл, содержащий рёбра всех трёх цветов?

### Формат входных данных

Первая строка содержит целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 10^6$ ). Далее следуют описания тестовых случаев.

Первая строка каждого описания содержит два целых числа  $n$  и  $m$  — количество вершин и рёбер в графе ( $1 \leq n, m \leq 10^6$ ). Каждая строка  $i$  из следующих  $m$  строк содержит три целых числа  $x_i$ ,  $y_i$  и  $z_i$ : это означает, что  $i$ -е ребро соединяет вершины  $x_i$  и  $y_i$  и имеет цвет  $z_i$ . Цвета пронумерованы так: 1 — красный, 2 — жёлтый, 3 — синий. Гарантируется, что граф связан, а также не содержит петель и кратных рёбер.

Общее количество вершин во всех тестовых случаях составляет не более  $10^6$ , общее количество рёбер во всех тестовых случаях составляет не более  $10^6$ .

### Формат выходных данных

Для каждого тестового случая выведите одну строку. Если есть цикл, содержащий рёбра всех трёх цветов, выведите «Yes». В противном случае выведите «No». Буквы можно выводить в любом регистре.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	Yes
3 3	No
1 2 3	
2 3 1	
1 3 2	
5 6	
1 2 1	
2 3 1	
1 3 2	
3 4 3	
3 5 3	
4 5 3	

## Задача J. Диапазоны множеств

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

*Это интерактивная задача.*

У вас есть  $n$  множеств целых чисел. Множества пронумерованы от 1 до  $n$ . Изначально все множества пустые. Ваша задача — выполнить  $q$  запросов следующих трёх типов.

1. «+  $\ell$   $r$   $x$ ». Добавить  $x$  во все множества с номерами от  $\ell$  до  $r$  включительно ( $1 \leq \ell \leq r \leq n$ ).
2. «-  $\ell$   $r$   $x$ ». Удалить  $x$  из всех множеств с номерами от  $\ell$  до  $r$  включительно ( $1 \leq \ell \leq r \leq n$ ).
3. «?  $k$ ». Вывести размер множества с номером  $k$  ( $1 \leq k \leq n$ ).

Множества ведут себя как обычные множества: если мы добавляем элемент, который уже есть, или удаляем элемент, которого уже нет, ничего не происходит.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$ : количество наборов и количество запросов соответственно ( $1 \leq n \leq 10^9$ ;  $0 \leq q \leq 10^5$ ). Следующие  $q$  строк содержат запросы в описанном выше формате ( $1 \leq x \leq q$  в запросах).

### Формат выходных данных

Для каждого запроса третьего типа ваша программа должна вывести ответ на отдельной строке. Гарантируется, что количество таких запросов не превосходит 10 000. После вывода каждого ответа не забывайте выводить перевод строки и очищать буфер вывода, иначе решение получит вердикт «Idleness Limit Exceeded». Очистить буфер можно, например, вызовом `fflush (stdout)` в C или `C++, System.out.flush ()` в Java, `flush (output)` в Pascal или `sys.stdout.flush ()` в Python.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
736 10	0
? 1	1
+ 1 5 1	2
+ 2 600 2	2
? 1	2
? 2	1
+ 1 6 2	
? 1	
? 2	
- 1 6 2	
? 4	

## Задача К. Целая полусумма

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

На доске написаны целые числа от  $\ell$  до  $r$  включительно, каждое — ровно один раз. За один шаг можно выбрать на доске два числа  $a$  и  $b$ , полусумма  $\frac{a+b}{2}$  которых — **целое** число, стереть с доски выбранные два числа и написать на доске их полусумму.

Может ли после нескольких (нуля или более) шагов на доске остаться ровно одно число? Если да, то какое максимальное число может при этом получиться?

### Формат входных данных

В первой строке заданы два целых числа  $\ell$  и  $r$  — минимальное и максимальное числа, изначально записанные на доске ( $1 \leq \ell \leq r \leq 100$ ).

### Формат выходных данных

Выведите максимальное число, которое может в одиночестве остаться на доске. Если на доске не может получиться одно число, выведите  $-1$ .

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
2 4	3	<u>2</u> , 3, <u>4</u> $\rightarrow$ <u>3</u> , 3 $\rightarrow$ 3

## Задача L. йыньливарпеН порядок

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

*Это задача с двойным запуском.*

У Вас есть массив из  $n \leq 1000$  беззнаковых 64-битных целых чисел. Вы собираетесь быстро переслать его на другой компьютер. Для этого вы параллельно отправляете каждое число через Интернет и потом снова собираете массив.

Однако возникла непредвиденная проблема. Как известно, «сетевой» порядок байт в многобайтовом числе отличается от используемого в современных компьютерах: если в современном компьютере байты записываются от младшего к старшему (little-endian), то при сетевом порядке байты записываются от старшего к младшему (big-endian). При преобразовании каждое число записывается как последовательность из 8 байт, и байты записываются в обратном порядке. И в некоторых случаях из-за сбоя на серверах обратное преобразование сделано не было...

То есть каждый элемент массива может прийти либо в обычном little-endian порядке, либо в сетевом big-endian. Порядок элементов массива сохранён. При передаче вы можете использовать не более 1024 64-битных чисел (то есть не более 8 **кибибайт**). Ваша задача — передать исходный массив и восстановить его после возможных искажений.

### Формат входных данных

Если требуется передать массив, то в первой строке записано слово «**encode**», во второй — целое число  $n$  ( $1 \leq n \leq 1000$ ), а в третьей —  $n$  целых чисел в интервале от 0 до  $2^{64} - 1$ .

Если требуется получить массив, то в первой строке записано слово «**decode**», во второй — целое число  $k$  ( $k \leq 1024$ ), а в третьей —  $k$  целых чисел в интервале от 0 до  $2^{64} - 1$ . Гарантируется, что числа идут в том же порядке, в котором они были переданы, и что любое из чисел либо передаётся неизменным, либо имеет перевёрнутый порядок байтов (в соответствии с условием задачи).

### Формат выходных данных

В случае передачи массива в первой строке выведите одно целое число  $k \leq 1024$ : количество передаваемых чисел. Во второй строке выведите  $k$  целых чисел в интервале от 0 до  $2^{64} - 1$ .

В случае получения массива выведите  $n$  целых чисел — исходный массив.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
encode 3 15 10 2023	6 15 15 10 10 2023 2023
decode 6 15 15 10 720575940379279360 16647274547598327808 2023	15 10 2023

### Замечание

В нижнем примере — в случае получения массива — все шесть чисел будут даны в одной строке. Дополнительный перевод строки добавлен только для удобства чтения.

На каждом тесте ваша программа будет запущена два раза: в первый раз на передачу массива, а во второй раз на получение, причём в качестве входных данных будет передан вывод первого запуска с возможными искажениями. Тест считается пройденным, если исходный массив восстановлен корректно.

## Задача М. Хорошие разбиения

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

На вещественной прямой расположено  $2n$  различных точек, пронумерованных от 1 до  $2n$ . Будем говорить, что способ разбить их на  $n$  пар  $(a_1, b_1), \dots, (a_n, b_n)$  *хороший*, если точки во всех парах можно соединить кривыми так, что все  $n$  кривых не пересекаются друг с другом и не пересекают вещественную прямую. Сколько существует хороших способов разбить точки на пары? Поскольку это число может быть большим, выведите его по модулю  $p$  — заданного простого числа.

Два способа разбиения считаются одинаковыми, если можно переставить пары и переставить точки в парах так, чтобы из одного способа получить другой.

### Формат входных данных

В единственной строке записаны два целых числа  $N$  и  $p$ : максимальное число пар точек и простой модуль ( $1 \leq N \leq 200$ ;  $10^8 < p < 10^9$ ).

### Формат выходных данных

Выведите  $N$  строк: ответ на задачу для  $n = 1, 2, \dots, N$ . Каждый ответ выведите по модулю  $p$ .

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 998244353	1 3 14 84 592

## Задача N. Ботинки

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Планируя свой отпуск, Никита решил провести его с пользой и купить себе новую пару обуви. Для этого, он изучил ассортимент магазинов, расположенных на той же улице, что и его отель, и выбрал в них  $n$  пар обуви для примерки. Никита знает, что каждая пара обуви требует  $k$  секунд для примерки, и планирует каждый день выделять по  $T$  секунд для похода по магазинам. Улица представляет из себя координатную прямую, скорость движения по которой составляет единицу в секунду, а в нуле расположен отель, где должен начинаться и заканчиваться каждый поход по магазинам. Найдите наименьшее количество дней отпуска, которые понадобятся Никите для того, чтобы гарантированно примерить всю интересующую его обувь.

### Формат входных данных

В первой строке входного файла даны три целых числа  $n$ ,  $k$  и  $T$  ( $1 \leq n \leq 10^4$ ;  $1 \leq k \leq T \leq 10^{18}$ ): количество пар обуви, время, требующееся на примерку, и количество ежедневного свободного времени, которое можно потратить на хождение по магазинам. Вторая строка содержит  $n$  целых чисел: координаты пар обуви, запланированных для примерки. Обратите внимание, что эти координаты не обязательно различны: некоторые пары обуви могут находиться в одном и том же магазине, а магазин может быть расположен на территории отеля. При этом гарантируется, что каждую пару обуви можно успеть примерить за один вечер.

### Формат выходных данных

Выведите одно целое число — ответ на задачу.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 100 400 -2 -1 -150 99 98	3