

## Задача А. Наперегонки

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

Борис очень любит животных, поэтому у него дома живут два прожорливых кота: Барсик и Персик. Любимое время дня котов — это время ужина, когда их хозяин Борис приходит с работы и накладывает в их миски корм. Коты начинают есть одновременно каждый из своей миски, но из-за своей жадности в какой-то момент обязательно меняются мисками. Борис знает, что больше всего в своей пушистой жизни Барсик и Персик любят поесть, а меньше всего — смотреть, как ест кто-то другой. Помогите Борису определить, в какой момент времени коты должны поменяться мисками, чтобы они закончили есть одновременно.

Миски Барсика и Персика имеют разные размеры и вмещают по  $n_1$  и  $n_2$  подушечек корма соответственно. Барсик поедает корм со скоростью  $v_1$  подушечек корма в минуту, а Персик — со скоростью  $v_2$  подушечек корма в минуту. Барсик и Персик начинают есть одновременно каждый из своей миски. Через некоторое целое количество минут они меняются мисками и продолжают есть из чужих мисок каждый со своей скоростью. При этом коты могут не съесть ни одной подушечки корма из своих мисок, но обязательно должны съесть хотя бы одну подушечку из чужой миски. Можно считать, что коты едят равномерно и непрерывно, а обмен мисками происходит мгновенно. Определите, через какое целое количество минут коты должны поменяться мисками, чтобы закончить есть одновременно. Обратите внимание, что количество минут, которое коты будут есть из чужих мисок может быть и не целым.

### Формат входных данных

На вход подается четыре строки. В каждой строке расположено одно целое число.

- В первых двух строках находятся числа  $n_1$  и  $n_2$  ( $1 \leq n_1, n_2 \leq 2 \cdot 10^9$ ) — количество подушечек корма в мисках Барсика и Персика соответственно.
- В последующих двух строках находятся числа  $v_1$  и  $v_2$  ( $1 \leq v_1, v_2 \leq 2 \cdot 10^9$ ) — количество подушечек корма, которые Барсик и Персик съедают за минуту.

### Формат выходных данных

Нужно вывести одно целое число — количество минут, по прошествии которого Барсик и Персик должны поменяться мисками. Если существует несколько подходящих ответов, нужно вывести максимальный из них. Если в указанных ограничениях невозможно поменяться мисками так, чтобы коты закончили есть одновременно — выведите  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
4 4 2 4	-1
4 5 2 4	1

### Замечание

Обратите внимание, что для работы с числами, не вмещающимися в 32-битный тип данных (превышающими  $2^{31} - 1$ ), в некоторых языках программирования нужно использовать специальный тип данных: `int64` в Pascal, `long long` в C++, `long` в Java и C#.

## Задача В. Самолетик

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

Посчитать площадь поверхности самолетика из бумаги в момент запуска, когда самолетик находится в руке создателя и плотно сжат посередине.

### Формат входных данных

В единственной строке находятся два целых числа  $a, b$  - размеры листа бумаги по горизонтали и вертикали соответственно ( $1 \leq a, b \leq 10^4$ ). Гарантируется, что  $b \geq a/2$ .

### Формат выходных данных

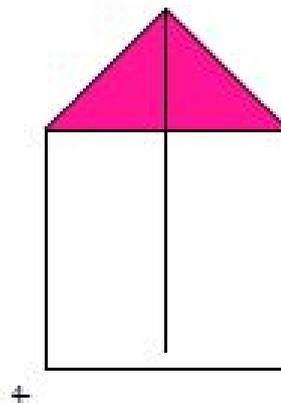
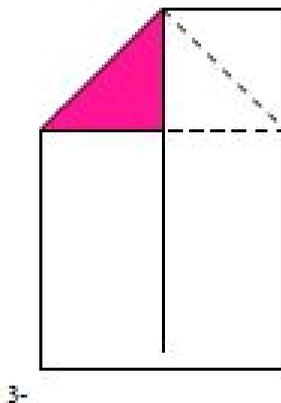
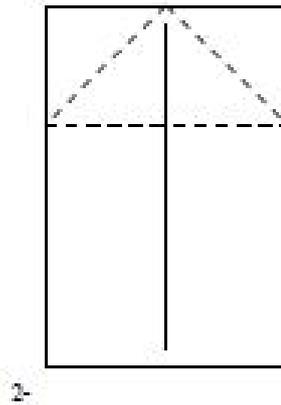
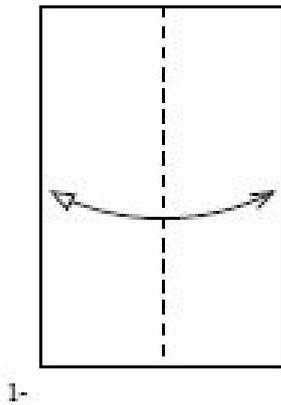
Выведите единственное число - площадь поверхности самолетика с 6 знаками после запятой.

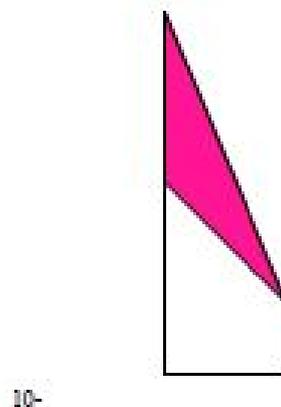
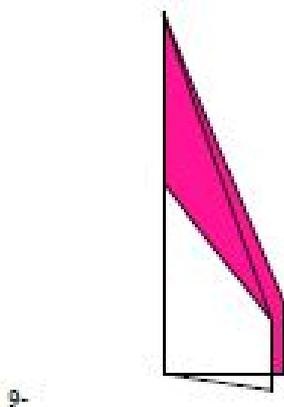
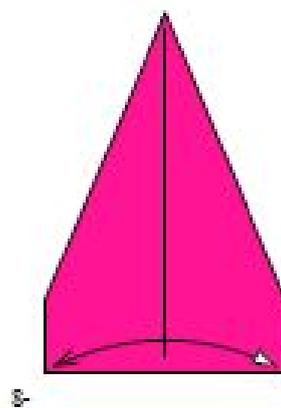
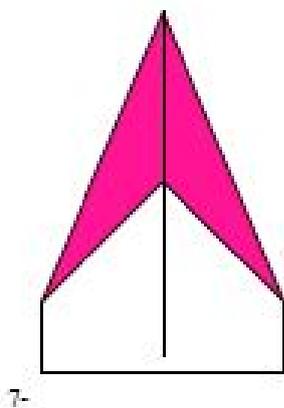
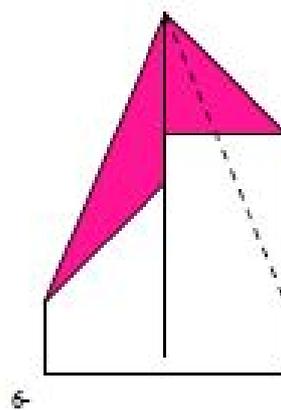
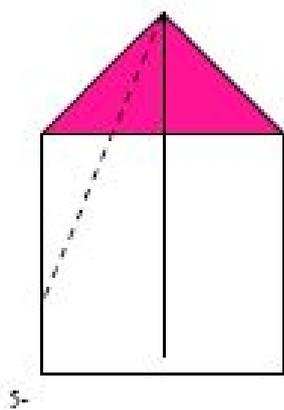
### Пример

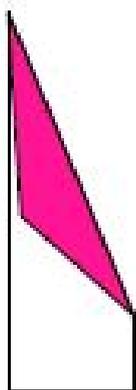
стандартный ввод	стандартный вывод
1 2	2.103553

### Замечание

Правила создания самолетика:







11-



12-

## Задача С. Короткая поездка

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

Павел решил провести следующие выходные в Москве. Однако, так как у него очень плотный график, он не может потратить на поездку больше, чем  $L$  а.е.в. Павел уже узнал расписание поездов и время их следования по маршруту Пермь-Москва. Помогите ему определить, какое максимальное количество времени он может рассматривать достопримечательности в столице.

Время задается не часами и минутами, а абстрактными единицами времени - а.е.в.

### Формат входных данных

В первой строке находятся два числа  $n, m$  ( $1 \leq n, m \leq 10^5$ ) - количества поездов из Перми в Москву и из Москвы в Пермь соответственно.

Во второй строке содержится  $n$  чисел ( $1 \leq a_i \leq 10^9$ ) - времена отправления поездов из Перми.

В третьей строке содержится  $m$  чисел ( $1 \leq b_i \leq 10^9$ ) - времена отправления поездов из Москвы.

В последней строке находится два числа  $L$  и  $T$  ( $1 \leq L, T \leq 10^9$ ) - ограничение по времени на всю поездку и время, которое тратит поезд для того, чтобы доехать из одного города в другой.

### Формат выходных данных

Выведите единственное число - максимальное возможное время в а.е.в., которое можно провести в Москве.

### Примеры

стандартный ввод	стандартный вывод
1 10 2 29 8 26 17 10 30 13 13 7 7 18 2	13
1 1 10 1 10 1	0

### Замечание

В первом наборе входных данных, максимально возможное время будет достигнуто, если Павел выедет из Перми на поезде, отправляющемся в момент 2, а из Москвы выедет в момент 17. Он будет в Москве с 4 по 16 момент включительно, то есть 13 а.е.в., на всю поездку он затратит ровно 17 а.е.в. Таким образом, в поезде он будет в моменты: 2-3, 17-18.

Во втором наборе, Павел вообще не может организовать свою поездку так, чтобы доехать до Москвы и вернуться.

## Задача D. Дружелюбные великаны

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

После кораблекрушения посреди Тихого океана, группу учёных вынесло на остров, который ранее считался необитаемым. Побродив по нему, группа обнаружила огромное поселение. Не желая быть съеденными хозяевами острова, учёные решили установить наблюдение за жителями города.

Исследователям удалось узнать, что в поселении живёт  $n$  дружелюбных великанов. Каждый из них дружит хотя бы с  $\lfloor \frac{n}{2} \rfloor$  другими гигантами. Естественно, если первый великан дружит со вторым, то и второй дружит с первым. К сожалению, у жителей этого острова очень переменчивое настроение, поэтому часто какие-то пары великанов, начинают или перестают дружить. Также ученые отметили, что каждый великан занимается своим ремеслом, поэтому каждой паре коренных жителей острова необходимо периодически взаимодействовать, но великан может обмениваться товарами только со своим другом. Поэтому, чтобы торговать со своим недругом ему потребуется использовать посредников.

К великой радости учёных, на момент начала исследований аборигенов удалось пронумеровать числами от 1 до  $n$ , причём таким образом, что если они встанут в круг, в котором у соседних великанов номера отличаются на 1 (первый великан стоит рядом с  $n$ -м), то у каждого коренного жителя острова  $m$  соседей слева и  $m$  справа будут его друзьями, а остальные не будут.

После долгих наблюдений у учёных появился список изменений статуса дружбы у великанов. Обработав данные, они решили каждому изменению сопоставить параметр  $k_i$ . Где  $k_i$  — наименьшее целое число такое, что любые два великана могут обмениваться товарами, используя не более  $k_i$  посредников, после  $i$  изменений для  $i$  от 0 до  $K$ .

Наблюдение за неизвестными созданиями, как и любой эксперимент, — очень неточный способ познания мира, поэтому некоторые записи могут быть неправильными, поэтому если новый статус пары великанов совпадает с прежним, то ничего не происходит.

### Формат входных данных

В первой строке три целых числа  $n, m, K$  ( $1 < n \leq 10^6, \frac{n}{4} < m \leq 10^6, 0 < K \leq 10^5$ ), количество великанов на острове, количество друзей у каждого из великанов в начале исследований, делённое на 2 и количество записей. Заметьте, что  $2m$  может быть больше, чем  $n$ , тогда считается, что каждый великан дружит с каждым.

В следующих  $K$  строках записи двух видов  $0 \ x \ y$  — великаны с индексами  $x, y$  поссорились (не гарантируется, что они до этого дружили) и  $1 \ x \ y$  — великаны с индексами  $x, y$  подружились (не гарантируется, что они до этого не дружили) ( $1 \leq x, y \leq n$ ).

### Формат выходных данных

В  $K + 1$  строке необходимо вывести коэффициент  $k_i$  для всех  $i$  от 0 до  $K$ . Если  $k_i = \infty$ , то необходимо вывести  $n$ .

### Пример

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

## Задача Е. Оля и перестановка

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

Недавно Оля узнала, что такое перестановка. *Перестановка размера  $n$*  — это упорядоченный набор без повторений чисел  $1, 2, \dots, n$ . Например, набор  $5, 2, 1, 3, 4$  является перестановкой, а наборы  $5, 2, 1, 3, 3$  и  $6, 2, 1, 3, 4$  не являются.

Учитель решил проверить, насколько Оля хорошо разбирается в перестановках. Он дал Оле массив  $a$  размера  $n$ , полученный с помощью перестановки  $p$  такого же размера  $n$  следующим образом:  $a_i$  равно позиции наименьшего элемента  $p_j$  такого, что  $p_j > p_i$  и  $j > i$  или 0, если такого элемента не существует. Иначе говоря  $a_i$  равно позиции наименьшего элемента, который больше чем  $p_i$  и который стоит справа от  $p_i$  в перестановке  $p$  или 0, если такого элемента нет.

Теперь учитель просит восстановить исходную перестановку. Оля справилась, а справитесь ли вы?

### Формат входных данных

В первой строке на вход подается одно натуральное число  $n$  ( $1 \leq n \leq 2000$ ) — размер перестановки и массива  $a$ . В следующей строке находятся  $n$  целых неотрицательных чисел  $a_i$  ( $0 \leq a_i \leq n$ ) — массив  $a$ , шифрующий перестановку. Гарантируется, что существует перестановка, соответствующая массиву  $a$ .

### Формат выходных данных

Выведите  $n$  натуральных чисел от 1 до  $n$ , описывающих искомую перестановку.

### Примеры

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