

Задача А. Дремуч лес

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Зайчик Ушастик живет в прекрасном лесу. У него есть две любимые поляны. Лес очень густой, поэтому солнце на этих полянах бывает нечасто. А именно: на первой поляне солнце светит каждую A -ю минуту, начиная с 0 -й, а на второй поляне каждую B -ю минуту, начиная с 0 -й.

Ушастик будет бегать между полянами ровно T минут в попытках увидеть любимое солнышко. От одной поляны до другой Ушастик добегаёт ровно за 1 минуту. Как только Заяц прибежит на одну поляну, то сразу же побежит на другую.

Сколько раз он увидит солнце? Ушастик стартует на первой поляне, при этом в самом начале на обеих полянах светит солнце.

Формат входных данных

Вводятся три целых числа A, B, T , каждое на отдельной строке. $1 \leq A, B \leq 2 \cdot 10^9, 0 \leq T \leq 2 \cdot 10^9$.

Формат выходных данных

Выведите одно целое число количество раз, когда Ушастик увидит солнце.

Система оценки

В задаче 20 тестов, не считая теста из условия. Каждый тест оценивается независимо в 5 баллов.

Пример

	стандартный ввод	стандартный вывод
4	5	
3		
10		

Замечание

Ушастик увидит солнце на первой поляне в моменты времени 0, 4 и 8, а на второй в моменты 3 и 9.

Задача В. Пончикова

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

Эта задача с открытыми тестами. Ее решением является набор ответов, а не программа на языке программирования. Тесты указаны в самом условии, от вас требуется лишь ввести ответы на них в тестирующую систему.

Ульяна открывает свою собственную пончиковую! Пока что у нее есть только один пончик, но, используя инновации и высокие технологии, Ульяна собрала особый Пончиковый автомат, который может совершать ровно одно действие — удваивать количество пончиков, которое есть на текущий момент.

Помимо использования автомата, у Ульяны всегда есть возможность самостоятельно приготовить еще один пончик.

Ульяна вычислила, что для открытия пончиковой ей необходимо получить ровно N пончиков. Помогите ей определить, какое наименьшее количество пончиков ей придется приготовить самостоятельно. Ульяна может чередовать использование автомата и самостоятельное приготовление пончиков в любом порядке.

Формат входных данных

Номер теста	Балл	N
1	10	5
2	10	29
3	10	1023
4	10	1000000
5	10	10000000
6	10	100000000
7	10	268435456
8	10	536870911
9	10	998496356
10	10	1000000000

Формат выходных данных

Выведите единственное число — минимальное количество самостоятельно приготовленных пончиков.

Примеры

стандартный ввод	стандартный вывод
6	1
7	2

Замечание

В первом примере из условия Ульяна может совершить три действия, чтобы получить $N = 6$ пончиков, при этом использовав автомат два раза:

1. Удвоить количество пончиков — получится 2.
2. Приготовить еще один пончик — получится 3.
3. Удвоить количество пончиков — получится 6.

Задача С. Умный Мост

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

Эта задача с открытым тестированием. Решения по этой задаче тестируются «в открытую», во время олимпиады. Вы можете не дожидаться окончания олимпиады, чтобы узнать итоговый балл вашего решения по этой задаче.

Вы разрабатываете новую технологию — Умный Мост (сокращенно УМ). УМ состоит из n сегментов, высота каждого сегмента регулируется отдельно. Основная задача УМа — подстраиваться под проходящие под мостом баржи. Баржи проходят одновременно, под каждой секцией моста — одна баржа.

Вам известна высота барж под каждым сегментом моста, а именно, под i -м сегментом высота баржи составляет a_i . УМ должен подстроиться под баржи, для этого нужно выбрать высоту каждого из n сегментов УМа. Обозначим выбранные высоты b_i . Должны выполняться два условия:

1. Каждый сегмент должен быть не ниже, чем высота баржи под ним, то есть $a_i \leq b_i$.
2. Каждый сегмент должен отличаться по высоте от соседних сегментов не более чем на 1. Формально, для всех $i \geq 2$ должно выполняться неравенство $|b_i - b_{i-1}| \leq 1$.

По известным высотам барж подберите такую конфигурацию моста (то есть последовательность чисел b_i), что сумма b_i минимальна.

Формат входных данных

В первой строке вводится единственное число n — количество сегментов моста ($1 \leq n \leq 200\,000$).

Во второй строке вводится n целых положительных чисел a_i — высоты барж под каждым сегментом моста ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите n целых положительных чисел b_i — оптимальную конфигурацию моста, которая удовлетворяет условиям и при этом имеет наименьшую сумму b_i .

Система оценки

Решения, корректно работающие для $n \leq 10^4$, будут набирать не менее 50 баллов.

Примеры

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

Замечание

В первом примере из условия, если под мостом из $n = 3$ сегментов проходят баржи с высотами $(3, 2, 4)$, то оптимальная конфигурация моста — это $(3, 3, 4)$. Заметьте, что нельзя выбрать высоту второго сегмента моста $b_2 = 2$, так как тогда разница с соседним сегментом номер 3 будет равна 2, что нарушает условие.

Задача D. Железнодорожный конструктор

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

Эта задача с открытым тестированием. Решения по этой задаче тестируются «в открытую», во время олимпиады. Вы можете не дожидаться окончания олимпиады, чтобы узнать итоговый балл вашего решения по этой задаче.

Олегу подарили на день рождения увлекательный железнодорожный конструктор. Он состоит из нескольких компонентов:

- Вагоны, пронумерованные от 1 до n . Каждый вагон уникален;
- Стартовый путь;
- Конечный путь;
- k сортировочных путей.

Стартовый и конечный пути соединены с каждым из сортировочных, а именно:

- Последний вагон на стартовом пути можно перегнать в любой из сортировочных, и, таким образом, он станет там первым. Все остальные вагоны на выбранном сортировочном пути сдвигаются.
- Последний вагон любого сортировочного пути можно перегнать на конечный путь, и, таким образом, он станет там первым. Все остальные вагоны, уже стоящие на конечном пути, сдвигаются.

Олег расставил все n вагонов на стартовом пути некоторым образом. Теперь его заинтересовал вопрос — получится ли у него отсортировать вагоны так, чтобы они все оказались на конечном пути в порядке от 1 до n ?

Формат входных данных

В первой строке вводится два целых числа n и k ($1 \leq k \leq 2\,000\,000$) — количество вагонов и количество сортировочных путей соответственно.

Во второй строке вводится n целых чисел от 1 до n — порядок вагонов на стартовом пути. Гарантируется, что каждое число от 1 до n встречается ровно один раз.

Формат выходных данных

Выведите YES, если у Олега получится отсортировать вагоны, выведите NO в противном случае.

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу ставятся при прохождении всех тестов этой группы, а также при прохождении тестов необходимых подгрупп. Баллы за соответствующие группы указаны в таблице.

Группа	Баллы	Дополнительные ограничения	Необходимые группы	Комментарий
0	0			Тесты из условия
1	30	$k \leq 2$	0	
2	20	$k \leq 3$	1	
3	50	Без дополнительных ограничений	1,2	

Примеры

стандартный ввод	стандартный вывод
5 1 1 2 3 4 5	YES
5 2 1 3 2 4 5	YES
5 2 5 4 3 2 1	NO

Замечание

Во втором тесте из условия порядок действий Олега может быть следующим:

1. Перегнать вагон 5 в первый сортировочный путь
2. Перегнать вагон 4 в первый сортировочный путь
3. Перегнать вагон 2 в первый сортировочный путь
4. Перегнать вагон 3 во второй сортировочный путь
5. Перегнать вагон 1 во второй сортировочный путь
6. Перегнать вагон 5 из первого сортировочного пути на конечный путь
7. Перегнать вагон 4 из первого сортировочного пути на конечный путь
8. Перегнать вагон 3 из второго сортировочного пути на конечный путь
9. Перегнать вагон 2 из первого сортировочного пути на конечный путь
10. Перегнать вагон 1 из второго сортировочного пути на конечный путь