

Задача А. Порядок

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

Программист Кеша с недавних пор занят в новом проекте. В связи с этим на его рабочем столе появились три новые книги. Кеша использует их в течение всего рабочего дня, а когда заканчивает работу, складывает их в стопку — одну на другую.

Известны высота, ширина и толщина каждой книги. Кеша заинтересовался, можно ли сказать, что книги упорядочены хотя бы по какой-то из этих величин? Ваша задача — определить это.

Формат входных данных

В первой строке содержатся через пробел три целых положительных числа h_1, w_1, t_1 — высота, ширина и толщина самой нижней в стопке книги.

Во второй строке содержатся через пробел три целых положительных числа h_2, w_2, t_2 — высота, ширина и толщина книги, которая лежит в середине стопки.

В третьей строке содержатся через пробел три целых положительных числа h_3, w_3, t_3 — высота, ширина и толщина самой верхней в стопке книги.

Гарантируется, что все числа попарно различны и не превосходят 100.

Формат выходных данных

Выведите строку из трёх символов $s_1s_2s_3$.

Если книги можно считать упорядоченными по высоте, $s_1 = H$, в противном случае $s_1 = N$.

Если книги можно считать упорядоченными по ширине, $s_2 = W$, в противном случае $s_2 = N$.

Если книги можно считать упорядоченными по толщине, $s_3 = T$, в противном случае $s_3 = N$.

Система оценки

В первой подзадаче применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах второй подзадачи осуществляется только, если пройдены все тесты первой подзадачи. В этой подзадаче также применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	2 (до 10)	книги упорядочены только по одной из величин	нет	полная
2	2 (до 90)	все возможные варианты	1	полная

Примеры

стандартный ввод	стандартный вывод
1 2 7 4 10 5 15 9 3	HNT
20 40 80 1 9 15 4 7 30	NWN

Задача В. Наивысший приоритет

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

Программист Кеша начинает свой рабочий день с просмотра почтового ящика. Сегодня утром в его почтовом ящике обнаружилось n писем.

Кеша читает письма по порядку, начиная с письма $\#1$ и заканчивая письмом $\#n$, и назначает каждому письму приоритет. Приоритет — это целое положительное число, чем оно меньше, тем более важным является письмо. Так, самый высокий приоритет имеет значение 1. Когда Кеша назначает новому письму приоритет, то он изменяет приоритет всех менее важных писем, прочитанных им до этого.

В частном случае, когда Кеша читает письмо, которое является более важным, чем все прочитанные ранее, он назначает читаемому письму приоритет 1, а приоритет всех ранее прочитанных писем увеличивается на 1.

Ваша задача — по окончательной расстановке приоритетов p_j определить, сколько писем в процессе расстановки получали приоритет 1.

Формальное описание алгоритма, используемого Кешей

Когда Кеша прочитывает письмо $\#j$, он может назначить ему любое значение приоритета из диапазона от 1 до j . Если он назначил письму $\#j$ некоторое значение приоритета k ($k < j$), это приводит к увеличению всех назначенных ранее приоритетов, больших или равных k , на 1. Таким образом, в момент после прочтения письма $\#j$ приоритеты всех прочитанных писем образуют перестановку чисел от 1 до j .

Пояснение к используемому Кешей алгоритму

Когда Кеша прочитает первое письмо, он назначит ему приоритет 1.

Когда Кеша прочитает второе письмо, он может назначить ему приоритет 2, если оно менее важно, чем первое (текущий приоритет писем будет 1, 2), или приоритет 1, если оно более важно (текущий приоритет писем будет 2, 1).

Когда Кеша прочитает третье письмо, он может назначить ему приоритет 1, 2 или 3 в зависимости от важности этого письма.

Так, Кеша назначит третьему письму приоритет 1, если оно важнее первого и второго писем. В таком случае текущий приоритет писем будет 2, 3, 1 (если был 1, 2 до этого) или 3, 2, 1 (если был 2, 1).

Если же третье письмо менее важно, чем письмо, имеющее текущий приоритет 1, но более важно, чем письмо, имеющее текущий приоритет 2, Кеша назначит третьему письму приоритет 2. В этом случае приоритет писем станет 1, 3, 2 (если был 1, 2) или 3, 1, 2 (если до этого был 2, 1)

Наконец, если третье письмо менее важно, чем первое и второе письма, то его текущим приоритетом станет 3. В таком случае приоритет писем будет 1, 2, 3 (если был 1, 2) или 2, 1, 3 (если был 2, 1).

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество писем, полученных Кешей.

Во второй строке содержится n целых чисел p_1, p_2, \dots, p_n , ($1 \leq p_i \leq n$, $i = 1, 2, \dots, n$) — окончательные приоритеты писем в порядке чтения их Кешей.

Гарантируется, что $p_i \neq p_j$ для $i \neq j$.

Формат выходных данных

Выведите единственное целое число — количество писем, которые в процессе расстановки приоритетов получали приоритет 1.

Система оценки

В первой подзадаче применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решений на тестах второй, третьей и четвертой подзадачах осуществляется только, если все тесты первой подзадачи были пройдены. В этих подзадачах применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решений на тестах пятой подзадачи осуществляется только, если все тесты первых четырех подзадач были пройдены. Баллы за пятую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	1 (до 9)	$1 \leq n \leq 3$	нет	полная
2	1 (до 21)	$4 \leq n \leq 10$	1	полная
3	1 (до 15)	$11 \leq n \leq 100$	1	полная
4	1 (до 15)	$101 \leq n \leq 1000$	1	полная
5	0 (40)	$1001 \leq n \leq 3 \cdot 10^5$ ограничения по времени: 1.5 с на тест	1, 2, 3, 4	первая ошибка

Пример

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

Замечание

Поясним приведённый пример.

Для начала приведём таблицу, в которой продемонстрируем, как менялись приоритеты всех писем на каждом шаге. Прочерк в графе означает, что письмо ещё не было прочитано.

Номер письма Номер шага	1	2	3	4	5	6	7
1	1	-	-	-	-	-	-
2	2	1	-	-	-	-	-
3	2	1	3	-	-	-	-
4	3	1	4	2	-	-	-
5	3	1	5	2	4	-	-
6	4	2	6	3	5	1	-
7	4	2	6	3	5	1	7

Видно, что на первом, втором и шестом шагах Кеша присваивал письмам приоритет 1.

Детальное пояснение примера

Кеша читает первое письмо с окончательным приоритетом 4. Оно на данный момент единственное, поэтому получает приоритет 1.

Затем Кеша читает второе письмо с итоговым приоритетом 2. Так как $2 < 4$, то Кеша посчитал письмо более важным и назначил ему приоритет 1. Приоритет всех писем 2, 1.

Следующее — третье — письмо в окончательной расстановке имеет приоритет 6. Оно менее важное, чем предыдущие, поэтому получает приоритет 3, а общий приоритет становится 2, 1, 3.

Четвёртое письмо в окончательной расстановке приоритетов имеет приоритет 3. Среди прочитанных только второе письмо важнее его, поэтому на данный момент четвертое письмо получает приоритет 2. Общий приоритет получается 3, 1, 4, 2.

Пятое письмо в окончательной расстановке приоритетов получило приоритет 5. Оно более важно, чем третье письмо, но менее важно, чем все остальные, поэтому оно получает приоритет 4, а итоговый приоритет становится равным 3, 1, 5, 2, 4.

Шестое письмо в окончательной расстановке приоритетов имеет приоритет 1. Как только Кеша прочитает это письмо, он увидит, что письмо имеет самый высокий приоритет среди всех прочитанных ранее. Поэтому Кеша назначит шестому письму приоритет 1, а общий приоритет примет вид 4, 2, 6, 3, 5, 1.

Наконец, седьмое письмо в окончательной расстановке приоритетов имеет приоритет 7, что не меняет уже сложившейся картины - приоритет всех писем становится равным 4, 2, 6, 3, 5, 1, 7 - именно те данные, которые даны в примере.

Таким образом, в процессе чтения Кеша будет назначать наивысший приоритет последовательно первому, второму и шестому письмам. Всего таких писем получается 3.

Задача С. Маски

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Программист Кеша с пониманием относится к требованиям носить маску в местах массового скопления людей. Поэтому у него при себе имеется рулон, от которого можно оторвать очередную одноразовую маску и надеть её, когда это потребуется.

В ближайшее время Кеша планирует посетить n мест массового скопления людей (магазины, аптеки, общественный транспорт), при этом он знает, что в месте $\#j$ он проведёт a_j единиц времени. Эти места Кеша посещает в порядке нумерации, менять этот порядок он не станет.

В инструкции по ношению масок сказано, что одну маску можно носить не более t единиц времени.

Кеша действует следующим образом. Пусть он носит очередную маску в течение z единиц времени, и ему предстоит посетить место, в котором он проведёт a_j единиц времени. Тогда, если $z + a_j \leq t$, он вновь наденет эту маску. В противном же случае он выбросит эту маску и наденет новую.

Впрочем, Кеша полагает, что более k раз одну и ту же маску надевать все же не стоит. Так что, даже если время позволяет, маску, которую он уже надевал k раз, использовать вновь Кеша не будет.

Ваша задача — определить, сколько масок потребуется Кеше, чтобы посетить все n мест массового скопления людей. Также укажите минимальное и максимальное время, в течение которого Кеша пользовался одной и той же маской.

Гарантируется, что время t всегда будет достаточным, чтобы посетить любое из мест массового скопления людей (иными словами, $a_j \leq t$ для всех значений j).

Формат входных данных

В первой строке содержатся целые числа n, t, k ($1 \leq n \leq 3 \cdot 10^5$, $1 \leq t \leq 10^9$, $1 \leq k \leq n$) — количество мест массового скопления людей, которые планирует посетить Кеша; время, в течение которого можно использовать одну и ту же маску; количество раз, которое Кеша считает допустимым надевать одну и ту же маску.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_j \leq t$, $j = 1, 2, \dots, n$), a_j — время, которое Кеша проведёт в месте $\#j$.

Формат выходных данных

Выведите количество масок, которые использует Кеша, а также минимальное и максимальное время, в течение которого Кеша будет использовать одну и ту же маску. Разделяйте числа пробелом или переводом строки.

Система оценки

В первой подзадаче применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах второй, третьей, четвёртой и пятой подзадач осуществляется только, если пройдены все тесты первой подзадачи. В этих подзадачах применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах шестой подзадачи осуществляется только, если пройдены все тесты всех предыдущих подзадач. В этой подзадаче применяется потестовая система оценки. Участнику сообщаются номера тестов внутри подзадачи, которые не были пройдены.

Проверка решения на тестах седьмой подзадачи осуществляется только, если пройдены все тесты всех предыдущих подзадач. В этой подзадаче применяется потестовая система оценки. Участнику сообщаются номера тестов внутри подзадачи, которые не были пройдены.

Проверка решения на тестах восьмой подзадачи осуществляется только, если пройдены все тесты всех предыдущих подзадач. Баллы за эту подзадачу начисляются только в случае прохождения всех

тестов этой подзадачи. Участнику сообщается либо номер первого теста внутри подзадачи, который не пройден, и результат проверки на этом тесте, либо что все тесты этой подзадачи пройдены.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	1 (до 5)	$k = 1, n = 3$	нет	полная
2	1 (до 10)	$k > 1, n = 3$	1	полная
3	2 (до 10)	$k = 2$	1	полная
4	2 (до 10)	$k = n$	1	полная
5	2 (до 10)	$a_j \geq t/2$	1	полная
6	1 (до 10)	$1 \leq n \leq 100$	1, 2, 3, 4, 5	полная
7	1 (до 10)	$101 \leq n \leq 3000$	1, 2, 3, 4, 5, 6	полная
8	0 (35)	$3001 \leq n \leq 3 \cdot 10^5$	1, 2, 3, 4, 5, 6, 7	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 20 2 14 7 13	2 14 20
10 30 3 10 18 12 5 7 6 26 2 4 11	5 6 28

Замечание

Поясним приведённые примеры.

В первом примере Кеша использует две маски: одну для посещения первого места массового скопления людей, вторую — для посещения второго и третьего мест массового скопления людей.

Минимальное время, в течение которого он будет использовать одну и ту же маску, составит 14 единиц времени (первая маска), максимальное время, в течение которого он будет использовать одну и ту же маску, составит 20 единиц времени (вторая маска).

Во втором примере Кеша использует пять масок.

Первую маску Кеша будет надевать дважды: сначала на 10 единиц времени, затем на 18. В сумме это составит 28 единиц времени. Поскольку в третьем месте массового скопления людей он планирует пробыть 12 единиц времени, он не сможет ещё раз надеть эту же маску, поскольку тогда суммарное время её использования превысит 30 единиц времени.

Вторую маску Кеша наденет трижды: сначала на 12 единиц времени, затем на 5 единиц времени и, наконец, на 7 единиц времени. Хотя суммарное время использования маски составит 24 единицы времени, Кеша выбросит её, поскольку полагает, что надевать маску более трёх раз не стоит.

Третью маску Кеша наденет один раз на 6 единиц времени. Поскольку далее ему нужно будет носить маску в течение 26 единиц времени, он не станет использовать третью маску, а возьмёт четвёртую.

Четвёртую маску Кеша наденет дважды: сначала на 26 единиц времени, затем на 2 единицы времени, что суммарно составит 28 единиц времени. Надевать эту же маску ещё на 4 единицы времени он не будет.

Пятую маску Кеша также наденет дважды: на 4 единицы времени и на 11 единиц времени, суммарно это составит 15 единиц времени.

Таким образом, Кеша использует пять масок, при этом минимальное время использования одной и той же маски составит 6 единиц времени (третья маска), а максимальное время использования одной и той же маски составит 28 единиц времени (первая и четвёртая маски).

Задача D. Корпоративный чат

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

Поскольку Кеша работает дистанционно, он следит за сообщениями, которые появляются в корпоративном чате. Но, поскольку он полагает, что слишком частое чтение сообщений в чате плохо сказывается на производительности труда, он читает корпоративный чат один раз в течение рабочего дня, а затем ещё раз в конце рабочего дня.

В конце рабочего дня Кеша также подсчитывает *суммарную задержку*, с которой он прочитал пришедшие в течение дня сообщения. Эта задержка равна сумме разностей между моментом, когда Кеша прочитал сообщение, и моментом, в который оно пришло. Ради простоты будем полагать, что Кеша читает сообщения мгновенно, ровно в тот момент, когда открывает корпоративный чат.

Сегодня Кеша заметил, что очень удачно выбирал время для чтения корпоративного чата. В конце рабочего дня он открыл чат одновременно с появлением там последнего сообщения. А момент для чтения, выбранный им в течение рабочего дня, был таким, что суммарная задержка оказалась минимально возможной. Ваша задача — определить, в какой момент Кеша решил прочитать сообщения в корпоративном чате в течение рабочего дня.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 5 \cdot 10^5$) — количество сообщений, которые пришли в течение дня в корпоративный чат.

Во второй строке содержится n целых чисел t_1, t_2, \dots, t_n ($0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq 10^9$), где t_j — момент времени, в который пришло сообщение $\#j$ ($j = 1, 2, \dots, n$).

Формат выходных данных

Выведите два целых числа: момент времени, в который Кеша читал сообщения в течение рабочего дня, а также минимально возможная суммарная задержка, с которой он прочитал пришедшие в течение дня сообщения.

Если существует несколько оптимальных вариантов ответа, выведите наиболее ранний момент времени, в который Кеша мог прочитать сообщения. Если Кеше не нужно было прерываться на чтение в течение рабочего дня для минимизации задержки, то выведите время чтения сообщений после работы.

Разделяйте числа пробелом или переводом строки.

Система оценки

В первой и второй подзадачах применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за подзадачу. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах третьей подзадачи осуществляется только, если пройдены все тесты второй подзадачи. В этой подзадаче также применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах четвёртой подзадачи осуществляется только, если пройдены все тесты всех предыдущих подзадач. В этой подзадаче также применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решения на тестах пятой подзадачи осуществляется только, если пройдены все тесты всех предыдущих подзадач. Баллы за пятую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого теста внутри подзадачи, который не пройден, и результат проверки на этом тесте, либо что все тесты этой подзадачи пройдены.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	1 (до 10)	$n = 5, t_i \leq 100$ для всех i	нет	полная
2	1 (до 10)	$2 \leq n \leq 100,$ $t_i \leq 10^4, t_i \neq t_j$ при $i \neq j$	нет	полная
3	1 (до 15)	$2 \leq n \leq 100, t_i \leq 10^4$	1, 2	полная
4	1 (до 25)	$101 \leq n \leq 1000, t_i \leq 10^9$	1, 2, 3	полная
5	0 (40)	$1001 \leq n \leq 5 \cdot 10^5, t_i \leq 10^9,$ ограничения по времени: 1.5 с на тест	1, 2, 3, 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
7 12 19 22 34 45 47 52	22 43
7 11 11 15 20 26 26 35	15 41
2 11 11	11 0

Замечание

Поясним приведённые примеры.

В первом примере Кеша откроет рабочий чат в момент 22, когда придёт третье сообщение, то суммарная задержка будет вычисляться следующим образом: $(22 - 12) + (22 - 19) + (22 - 22) + (52 - 34) + (52 - 45) + (52 - 47) = 43$. Можно убедиться, что это наименьшее значение среди всех вариантов.

Например, если Кеша откроет рабочий чат в момент 10, он не обнаружит там ни одного сообщения, и, когда он откроет рабочий чат в момент 52, то суммарная задержка для сообщений составит $(52 - 12) + (52 - 19) + (52 - 22) + (52 - 34) + (52 - 45) + (52 - 47) = 133$.

Если Кеша откроет рабочий чат в момент 12, то он без задержки прочитает самое первое сообщение, а суммарная задержка составит $(12 - 12) + (52 - 19) + (52 - 22) + (52 - 34) + (52 - 45) + (52 - 47) = 93$.

Если же Кеша откроет рабочий чат в момент 35, то он прочитает в этот момент первые четыре сообщения, так что суммарная задержка будет вычисляться так: $(35 - 12) + (35 - 19) + (35 - 22) + (35 - 34) + (52 - 45) + (52 - 47) = 65$.

Во втором примере Кеша мог минимизировать суммарную задержку в двух случаях: когда он прочитал пришедшие сообщения в момент времени 15 и когда он прочитал пришедшие сообщения в момент 20. В обоих случаях эта суммарная задержка составит 41 единицу времени, но следует вывести наиболее ранний момент времени, поэтому в ответе 15.

В третьем примере Кеша получает два сообщения в конце рабочего дня. Именно в этот момент он их и прочитает.

Задача Е. И снова о приоритетах

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

Когда Кеша едет в трамвае или идёт пешком знакомым маршрутом, он имеет привычку размышлять над какой-нибудь задачей. Вот и сейчас он задумался о процессе просмотра своего почтового ящика, описание которого вы могли прочесть в задаче В. Напомним, как это происходит.

Кеша читает письма по порядку, начиная с письма #1 и заканчивая письмом # n , и назначает каждому письму приоритет. Приоритет — это целое положительное число, чем оно меньше, тем более важным является письмо. Так, самый высокий приоритет имеет значение 1. Когда Кеша назначает новому письму приоритет, то он изменяет приоритет всех менее важных писем, прочитанных им до этого.

Для каждого из n писем известны приоритеты q_j , которые Кеша назначал им сразу после прочтения. Ваша задача — определить итоговую расстановку приоритетов после прочтения всех писем.

Формальное описание алгоритма, используемого Кешей

Когда Кеша прочитывает письмо # j , он может назначить ему любое значение приоритета из диапазона от 1 до j . Если он назначил письму # j некоторое значение приоритета k ($k < j$), это приводит к увеличению всех назначенных ранее приоритетов, больших или равных k , на 1. Таким образом, в момент после прочтения письма # j приоритеты всех прочитанных писем образуют перестановку чисел от 1 до j .

Пояснение к используемому Кешей алгоритму

Когда Кеша прочитает первое письмо, он назначит ему приоритет 1.

Когда Кеша прочитает второе письмо, он может назначить ему приоритет 2, если оно менее важное, чем первое (текущий приоритет писем будет 1, 2), или приоритет 1, если оно более важное (текущий приоритет писем будет 2, 1).

Когда Кеша прочитает третье письмо, он может назначить ему приоритет 1, 2 или 3 в зависимости от важности этого письма.

Так, Кеша назначит третьему письму приоритет 1, если оно важнее первого и второго писем. В таком случае текущий приоритет писем будет 2, 3, 1 (если был 1, 2 до этого) или 3, 2, 1 (если был 2, 1).

Если же третье письмо менее важное, чем письмо, имеющее текущий приоритет 1, но более важное, чем письмо, имеющее текущий приоритет 2, Кеша назначит третьему письму приоритет 2. В этом случае приоритет писем станет 1, 3, 2 (если был 1, 2) или 3, 1, 2 (если до этого был 2, 1)

Наконец, если третье письмо менее важное, чем первое и второе письма, то его текущим приоритетом станет 3. В таком случае приоритет писем будет 1, 2, 3 (если был 1, 2) или 2, 1, 3 (если был 2, 1).

Формат входных данных

В первой строке содержится целое число n ($1 \leq n \leq 3 \cdot 10^5$) — количество писем в почтовом ящике.

Во второй строке содержится n целых чисел q_1, q_2, \dots, q_n ($1 \leq q_j \leq j$, $j = 1, 2, \dots, n$) через пробел. Число q_j — приоритет, который назначил Кеша письму # j сразу после его прочтения.

Формат выходных данных

Выведите n целых чисел. Число # j — итоговый приоритет письма # j . Разделяйте числа пробелами или переводами строк.

Система оценки

В первой подзадаче применяется потестовая система оценки. В графе «Баллы» указано количество баллов за тест и в скобках максимальное количество баллов, которое можно набрать за

подзадачу. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решений на тестах второй подзадачи осуществляется только, если все тесты первой подзадачи были пройдены. В этой подзадаче применяется потестовая система оценки. Участнику сообщаются номера тестов внутри этой подзадачи, которые не были пройдены.

Проверка решений на тестах третьей подзадачи осуществляется только, если все тесты первой и второй подзадач были пройдены. Баллы за третью подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Проверка решений на тестах четвёртой подзадачи осуществляется только, если все тесты первых трёх подзадач были пройдены. Баллы за четвёртую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Проверка решений на тестах пятой подзадачи осуществляется только, если все тесты первой и второй подзадач были пройдены. Баллы за пятую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Проверка решений на тестах шестой подзадачи осуществляется только, если все тесты первой, второй и пятой подзадач были пройдены. Баллы за шестую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Проверка решений на тестах седьмой подзадачи осуществляется только, если все тесты первых шести подзадач были пройдены. Баллы за седьмую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Проверка решений на тестах восьмой подзадачи осуществляется только, если все тесты первых семи подзадач были пройдены. Баллы за восьмую подзадачу начисляются только в случае прохождения всех тестов этой подзадачи. Участнику сообщается либо номер первого непройденного теста и результат проверки на этом тесте, либо что все тесты подзадачи пройдены.

Подзадача	Баллы за тест (баллы за подзадачу)	Ограничения	Необходимые подзадачи	Информация о проверке
1	1 (до 9)	$1 \leq n \leq 3$	нет	полная
2	1 (до 11)	$4 \leq n \leq 1000$	1	полная
3	0 (10)	$1001 \leq n \leq 10^5, j - 2 \leq q_j$	1, 2	первая ошибка
4	0 (10)	$1001 \leq n \leq 10^5, j - 49 \leq q_j$	1, 2, 3	первая ошибка
5	0 (10)	$1001 \leq n \leq 10^5, q_j \leq 3$	1, 2	первая ошибка
6	0 (10)	$1001 \leq n \leq 10^5, q_j \leq 50$	1, 2, 5	первая ошибка
7	0 (20)	$1001 \leq n \leq 10^5$	1, 2, 3, 4, 5, 6	первая ошибка
8	0 (20)	$10^5 + 1 \leq n \leq 3 \cdot 10^5$	1, 2, 3, 4, 5, 6, 7	первая ошибка

Пример

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

Замечание

Поясним приведённый пример.

Покажем, как менялись приоритеты писем на каждом шаге. Прочерк в графе означает, что письмо ещё не было прочитано.

Номер письма Номер шага	1	2	3	4	5	6	7
1	1	-	-	-	-	-	-
2	2	1	-	-	-	-	-
3	2	1	3	-	-	-	-
4	3	1	4	2	-	-	-
5	3	1	4	2	5	-	-
6	4	2	5	3	6	1	-
7	4	2	5	3	6	1	7