

## Задача А. Олимпиада

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

Маленький мальчик Гриша уже сам начал делать олимпиады, и ему как раз нужно подготовить Открытую Олимпиаду по Информатике. Для олимпиады нужно придумывать задачи, а Гриша как раз очень любит этим заниматься. Каждую задачу он характеризует ее сложностью, которая является неотрицательным целым числом. Сложность олимпиады Гриша оценивает как сумму сложностей всех задач в ней.

На Открытую Олимпиаду по Информатике Грише требуется придумать  $n$  задач. Конечно, Гриша еще не дорос до того, чтобы составлять олимпиаду самому, поэтому за его действиями следит многоуважаемое начальство. Начальство любит отвергать некоторые задачи, но Гриша знает, что оно не может отклонить больше, чем  $k$  из его задач. Также Гриша, исходя из своего большого опыта, считает, что давать олимпиаду со сложностью меньше чем  $x$  будет бессмысленно, поэтому он не допустит такого.

Как мы уже говорили, Гриша — настоящий специалист в придумывании задач и может придумать сколько угодно задач любой сложности. Он хочет придумать  $n$  задач, понимая, что часть из них могут отклонить, и, конечно, он хочет, чтобы независимо от того, какие задачи его начальство отвергнет, сложность олимпиады из оставшихся задач была не меньше  $x$ . Конечно, Гриша является еще и лентяем, поэтому не хочет перетруждаться и планирует придумать  $n$  задач с минимальной суммарной сложностью. К сожалению, сейчас он занят учебой и не может рассчитать минимальную суммарную сложность этих задач. Помогите Грише, ведь до олимпиады осталось не так много времени.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $k$  и  $x$  ( $2 \leq n \leq 10^9$ ,  $1 \leq k < n$ ,  $1 \leq x \leq 10^9$ ) — количество задач, которое хочет придумать Гриша, максимальное количество задач, которое может отвергнуть его начальство и минимальная допустимая сложность конкурса по мнению Гриши.

### Формат выходных данных

Выведите одно целое число — минимальную суммарную сложность придуманных Гришей задач.

### Примеры

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

### Замечание

В первом тесте из условия Гриша должен придумать три задачи, при этом начальство может отклонить одну из них, а сложность олимпиады должна быть не меньше 5. В таком случае Гриша может придумать две задачи сложности 3 и одну задачу сложности 2. Тогда, какую бы задачу ни отвергнуло начальство, сложность конкурса будет равна 5 или 6, а суммарная сложность всех задач, придуманных Гришей, будет равна 8.

Во втором тесте из условия Гриша может придумать все три требуемые задачи со сложностью 1 и, какие бы две жюри ни выкинуло, оставшийся конкурс будет иметь сложность не менее 1. В таком случае суммарная сложность задач, придуманных Гришей будет равна 3.

### Система оценки

Тесты к этой задаче состоят из пяти подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$x$		
0	0	—	—	—	Тесты из условия.
1	13	$n \leq 3$	$x \leq 100$	0	
2	21	$n \leq 100$	$x \leq 100$	0 – 1	
3	24	$n \leq 10^6$	$x \leq 10^6$	0 – 2	
4	11	—	—	—	$x$ делится на $(n - k)$ .
5	31	—	—	0 – 4	

## Задача В. Дети и буквы

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

Алиса и Боб хотят пройти через турникет в метро. У них есть полоска, на которой в ряд написаны  $n$  строчных букв латинского алфавита. Для того, чтобы пройти через турникет, каждый должен приложить к нему бумажку с кодовым словом. Турникет сканирует слово, записанное на бумажке и открывается, позволяя пройти одному человеку. Чтобы запретить человеку ездить по одному билету несколько раз, турникет не откроется если приложить к нему бумажку со словом, которое турникет сканировал до этого.

Алиса и Боб пришли в метро к открытию, поэтому через турникет еще никто не проходил, и он пропустит человека с любым текстом на билете. Детям разрешено вырезать ровно по одному непустому билету из полоски, причем билеты должны состоять из подряд идущих букв полоски и **не должны пересекаться**. Также, чтобы никто из детей не считал себя обделенным, билеты, которые они вырежут, должны быть одинаковой длины. После вырезания билетов Алиса, а затем Боб попробуют пройти через турникет. Таким образом, Алиса сможет войти в метро всегда, а Боб сможет войти только если текст на его билете не совпадает с текстом на билете Алисы.

Так как дети маленькие, нужно помочь им определить, можно ли вырезать два различных билета одинаковой длины из полоски, чтобы попасть в метро, и, если это возможно, объяснить им, как это сделать.

### Формат входных данных

В первой строке задано одно целое число  $n$  ( $1 \leq n \leq 10^6$ ) — количество букв, написанных на полоске.

Во второй строке содержится строка, состоящая из  $n$  строчных букв латинского алфавита — текст на полоске.

### Формат выходных данных

Если невозможно вырезать из полоски два различных слова одинаковой длины выведите «No» (без кавычек).

Иначе в первой строке выведите «Yes» (без кавычек), а во второй строке выведите три целых числа — номер символа на полоске, с которого начнется билет Алисы, номер символа на полоске, с которого начнется билет Боба и длину билетов, которые вырежут дети. Каждый символ полоски должен попасть не более, чем в один билет, а также тексты на билетах должны быть различны. Символы на полоске нумеруются слева направо, **начиная с 1**.

Обратите внимание, что после вырезания билета из полоски, его нельзя переворачивать. Так, например, если вырезать билет `po` из полоски `pro`, то получить билет `od`, перевернув билет `po` сверху вниз, не получится.

### Примеры

стандартный ввод	стандартный вывод
8 abacabad	Yes 2 6 3
1 z	No

### Замечание

В первом примере Алиса вырежет билет, начинающийся со второго символа полоски и имеющий длину 3. Таким образом, на ее билете будет написано `bac`. Боб вырежет билет, начинающийся с шестого символа полоски и имеющий длину 3, поэтому на билете Боба будет написано `bad`.

Во втором примере полоска состоит только из одного символа, поэтому Алиса и Боб не смогут вырезать из нее два непустых слова так, чтобы ни один символ не попал в оба билета.

## Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения	Комментарий
		$n$	
0	0	–	Тесты из условия.
1	31	$n \leq 50$	
2	14	$n \leq 500$	
3	17	$n \leq 5000$	
4	28	$n \leq 100\,000$	<b>Offline-проверка.</b>
5	10	$n \leq 10^6$	<b>Offline-проверка.</b>

## Задача С. Метро

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

Пафнутий решил использовать метро, чтобы достичь своей цели.

Чтобы пройти через турникет, надо использовать специальную карточку SLON (Subterranean Lowcost Orientational Network). У каждой карточки есть неотрицательный целый баланс — количество бурлей, которые записаны на карточке. Если на карточке есть хотя бы  $k$  бурлей, то пройти через турникет возможно, после чего баланс карточки уменьшается на  $k$  бурлей. Если же на карточке меньше  $k$  бурлей, то Пафнутий не сможет ей воспользоваться, чтобы пройти через турникет.

Изначально у Пафнутия есть  $n$  карточек, баланс карточки с номером  $i$  составляет  $a_i$  бурлей. Также Пафнутий накопил  $p$  бурлей и может как угодно распределить эти деньги между карточками. Формально, пусть Пафнутий к балансу карточки с номером  $i$  добавил  $add_i \geq 0$  бурлей, тогда должно выполняться условие  $add_1 + add_2 + \dots + add_n \leq p$ . Возможности перераспределять деньги между карточками нет, то есть баланс карточки может увеличиваться только за счет какой-то части из этих  $p$  бурлей и уменьшаться только при проходе через турникет.

Пафнутий не очень любит математику. Помогите ему определить, какое максимальное количество раз он сможет поехать на метро, если распределит  $p$  бурлей по карточкам оптимально.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $p$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $0 \leq p \leq 10^{18}$ ,  $1 \leq k \leq 10^9$ ) — количество карточек, накопленная сумма и плата за один проход через турникет, соответственно.

Во второй строке задано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ ) — балансы карточек.

### Формат выходных данных

Выведите одно целое число — максимальное количество раз, которое Пафнутий сможет поехать на метро.

### Примеры

стандартный ввод	стандартный вывод
5 1 4 7 7 3 4 1	4
2 1000 2000 1299 1701	2

### Замечание

В первом примере из условия изначально можно поехать только 3 раза. Но если добавить 1 бурль на первую, вторую или третью карточку, то Пафнутий сможет поехать на метро 4 раза.

Во втором примере 1000 бурлей хватит только для того, чтобы сделать баланс обеих карточек равным 2000 бурлям.

### Система оценки

Тесты к этой задаче состоят из пяти подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$p$		
0	0	–	–	–	Тесты из условия.
1	13	–	–	–	$a_i$ делится на $k$ .
2	11	$n \leq 3$	$p \leq 100$	0	
3	28	$n \leq 100$	$p \leq 100$	0, 2	
4	23	$n \leq 1000$	–	0, 2 – 3	
5	25	–	–	0 – 4	<b>Offline-проверка.</b>

## Задача D. Дима и массив

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 6.5 секунд  
Ограничение по памяти: 768 мегабайт

Диме не дарили массив  $a$ , состоящий из  $n$  целых чисел на день рождения, он не покупал его, не находил на улице, а он у него просто есть и всегда был, и Диме не очень-то и интересно откуда.

Дима не играет с массивом, не дарит его Пете, не режет на кусочки и не стремится его уничтожить. Дима просто выполняет операции двух видов со своим массивом:

- $? \ l \ r$  — узнать **MEX** мультимножества  $\{a_l, a_{l+1}, \dots, a_r\}$
- $! \ i \ x$  — присвоить  $a_i$  значение  $x$  ( $0 \leq x \leq n$ )

**MEX** мультимножества чисел  $\{a_1, a_2, \dots, a_k\}$  — это минимальное целое  $t \geq 0$  такое, что  $t \neq a_i$  для всех  $1 \leq i \leq k$ .

На самом деле, Диме не очень нравится выполнять операции двух видов со своим массивом. Диму волнуют лишь результаты операций первого типа. Помогите Диме и напишите программу, которая выполнит операции за него.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $1 \leq n \leq 500\,000, 1 \leq q \leq 250\,000$ ) — размер массива, который есть у Димы и количество операций, соответственно.

Вторая строка содержит  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq n$ ) — массив Димы до начала операций.

Каждая из следующих  $q$  строк содержит описание одной операции в формате, описанном выше.

Гарантируется, что суммарно Дима сделал **не более 50 000 операций изменения массива**.

Элементы массива пронумерованы, начиная с 1.

### Формат выходных данных

Для каждой операция первого типа выведите одно целое число — **MEX** соответствующего мультимножества. Ответы на запросы выводите в порядке, в котором они заданы во входных данных.

### Пример

стандартный ввод	стандартный вывод
6 8	5
4 1 0 2 2 3	0
? 1 6	3
? 4 6	4
? 2 5	0
? 2 6	5
! 3 3	
? 1 6	
! 4 0	
? 1 6	

### Замечание

В примере запросы выглядят следующим образом:

- Изначально массив равен  $[4, 1, 0, 2, 2, 3]$
- В первом запросе считается **MEX** от  $\{4, 1, 0, 2, 2, 3\}$  и он равен 5.
- Во втором запросе считается **MEX** от  $\{2, 2, 3\}$  и он равен 0.
- В третьем запросе считается **MEX** от  $\{1, 0, 2, 2\}$  и он равен 3.

- В четвёртом запросе считается МЕХ от  $\{1, 0, 2, 2, 3\}$  и он равен 4.
- Пятый запрос меняет массив. Теперь он равен  $[4, 1, 3, 2, 2, 3]$ .
- В шестом запросе считается МЕХ от всего массива и он равен 0.
- Седьмой запрос меняет массив. Теперь он равен  $[4, 1, 3, 0, 2, 3]$ .
- В восьмом запросе снова считается МЕХ от всего массива и теперь он равен 5.

## Система оценки

Тесты к этой задаче состоят из тринадцати подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		$n$	$q$	$a_i, x$		
0	0	—	—	—	—	Тесты из условия.
1	11	$n \leq 100$	$q \leq 100$	—	0	
2	8	$n \leq 5000$	$q \leq 5000$	—	0, 1	
3	12	—	—	$a_i, x \leq 10$	0	
4	19	—	—	—	—	Нет операций изменения.
5	5	$n \leq 100\,000$	$q \leq 100\,000$	—	0 – 2	
6	5	$n \leq 150\,000$	$q \leq 150\,000$	—	0 – 2, 5	
7	5	$n \leq 200\,000$	$q \leq 200\,000$	—	0 – 2, 5, 6	
8	5	$n \leq 250\,000$	—	—	0 – 2, 5 – 7	
9	6	$n \leq 300\,000$	—	—	0 – 2, 5 – 8	Offline-проверка.
10	6	$n \leq 350\,000$	—	—	0 – 2, 5 – 9	Offline-проверка.
11	6	$n \leq 400\,000$	—	—	0 – 2, 5 – 10	Offline-проверка.
12	6	$n \leq 450\,000$	—	—	0 – 2, 5 – 11	Offline-проверка.
13	6	—	—	—	0 – 12	Offline-проверка.



## Задача Е. Петя и монеты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	512 мегабайт

Филателист Петя отправился в магазин, чтобы приобрести его заветную мечту — настоящую медную монету XVII века. В магазине продавец показал Пете  $n$  монет XVII века, имеющихся в наличии, каждая из которых сделана из одного из трех материалов: железа, меди или бронзы. К сожалению, время не пощадило монеты, поэтому они покрылись ржавчиной, и невозможно определить, какая монета из какого материала сделана. Чтобы Пете было проще найти нужную монету, продавец показал ему  $m$  пар монет таких, что монеты из пары сделаны из различных материалов. Кроме того, продавец сказал Пете, что ровно одна из  $n$  монет сделана из меди. Чтобы заполучить заветную монету, Петя решил приобрести все монеты, которые могут быть медными, исходя из информации, полученной им от продавца.

Более формально, Петя купит монету с номером  $p$  если и только если могло оказаться так, что все монеты с номерами, отличными от  $p$  сделаны из железа и бронзы, а монета с номером  $p$  сделана из меди и при этом в каждой из  $m$  пар номеров монет, озвученных продавцом, монеты с соответствующими номерами сделаны из разных материалов.

Определите, какие монеты купит Петя.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq n \leq 3 \cdot 10^6, 0 \leq m \leq 3 \cdot 10^6$ ) — количество монет XVII века в магазине и количество пар монет, про которые продавец сказал Пете, что они сделаны из разных материалов.

В следующих  $m$  строках заданы пары целых чисел  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ) — номера монет, которые сделаны из различных материалов. Все пары монет во вводе различны.

### Формат выходных данных

Если продавец ошибся и ни одна монета не может быть медной, выведите «0» (без кавычек). Иначе в первой строке выведите одно целое число  $x$  — количество монет, которые купит Петя.

Во второй строке выведите  $x$  целых чисел  $c_i$  ( $1 \leq c_i \leq n$ ) в порядке возрастания — номера монет, которые купит Петя.

## Примеры

стандартный ввод	стандартный вывод
4 3 1 2 1 4 4 2	3 1 2 4
2 1 1 2	2 1 2
6 6 1 2 4 5 2 3 5 6 1 3 4 6	0
12 14 1 2 3 2 3 4 5 4 5 6 6 7 3 8 8 9 9 6 2 10 10 12 12 11 11 7 1 7	4 2 3 6 7

## Замечание

В первом примере:

- Монета с номером 1 может быть медной, если, например, монеты с номерами 2 и 3 железные, а монета с номером 4 бронзовая.
- Монета с номером 2 может быть медной, если, например, монеты с номерами 1 и 3 железные, а монета с номером 4 бронзовая.
- Монета с номером 3 не может быть медной, так как среди монет с номерами 1, 2 и 4 не могут быть только железные и бронзовые.
- Монета с номером 4 может быть медной, если, например, монеты с номерами 2 и 3 железные, а монета с номером 1 бронзовая.

Во втором примере любая монета может быть медной, в случае, если, например, другая монета сделана из железа.

В третьем примере продавец ошибся и ни одна монета не может быть медной.

## Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Баллы	Доп. ограничения		Комментарий
		$n$	$m$	
0	0	–	–	Тесты из условия.
1	9	$n \leq 15$	$m \leq 100$	
2	11	$n \leq 5000$	$m \leq 5000$	
3	21	$n \leq 100\,000$	$m \leq 100\,000$	
4	28	$n \leq 10^6$	$m \leq 10^6$	
5	31	–	–	

## Задача F. Грустные танцы

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

Во Флатляндии проводится ежегодный турнир по танцам!

Из города  $NN$  приехала команда, состоящая из  $n$  танцоров, и вот настал день соревнований.

Состязания проходят в таком формате: танцоры пронумерованы от 1 до  $n$ , и изначально  $i$ -й танцор стоит на  $i$ -м месте. После этого они начинают танцевать по заранее согласованной программе выступления  $a$ : каждую минуту танцор с  $a_i$ -го места передвигается на  $i$ -е место, при этом все  $a_i$  различны. От команды требуется выстроиться так, чтобы  $i$ -й танцор оказался на  $b_i$ -м месте (аналогично, все  $b_i$  различны). После этого выступление завершается, и жюри оценивает его техничность и артистизм. При этом выступление должно продлиться хотя бы одну минуту, иначе оценивать будет просто нечего.

Но в этом году участники заподозрили жюри в подлоге: к ним пришла мысль, что, возможно, следуя программе  $a$ , они никогда не смогут занять требуемое положение  $b$ , что приводит к автоматическому поражению в турнире.

Так как они не программисты по образованию, команда города  $NN$  решила обратиться к вам за помощью: проверьте по их программе выступления  $a$  и требуемому положению  $b$ , существует ли такое положительное количество минут  $k$ , что через  $k$  минут после начала выступления  $i$ -й танцор будет находиться на  $b_i$ -м месте.

### Формат входных данных

В первой строке входного файла содержится целое число  $n$  ( $1 \leq n \leq 10^6$ ) — количество участников команды, приехавшей из города  $NN$ .

Вторая строка содержит  $n$  целых чисел  $a_1, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — программу выступления  $a$ . Гарантируется, что каждое число от 1 до  $n$  встречается в  $a$  ровно один раз.

Третья строка содержит  $n$  целых чисел  $b_1, \dots, b_n$  ( $1 \leq b_i \leq n$ ) — требуемое положение  $b$ . Гарантируется, что каждое число от 1 до  $n$  встречается в  $b$  ровно один раз.

### Формат выходных данных

Для каждого тестового примера выведите «Yes» (без кавычек), если существует такое количество минут  $k$ , что спустя  $k$  минут после начала выступления все танцоры будут в требуемом от них положении, или «No» (без кавычек), если такого  $k$  не существует.

### Примеры

стандартный ввод	стандартный вывод
4 2 3 4 1 1 2 3 4	Yes
4 1 2 3 4 2 1 4 3	No

### Замечание

В первом примере в нулевой момент времени танцоры располагаются так: 1 2 3 4. Но так как выступление должно продлиться хотя бы одну минуту,  $k = 0$  не подходит. Далее происходят следующие перемещения:

- 2 3 4 1 после первой минуты
- 3 4 1 2 после второй минуты
- 4 1 2 3 после третьей минуты

- 1 2 3 4 после четвертой минуты

Как видно, после четвертой минуты танцоры заняли требуемое положение, а значит, подходит  $k = 4$ , и ответ — «Yes».

Во втором примере танцоры всегда остаются на своем месте, следовательно, они никогда не займут требуемое положение, и ответ — «No».

## Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Баллы	Доп. ограничения	Комментарий
		$n$	
0	0	–	Тесты из условия.
1	10	$n \leq 8$	
2	40	$n \leq 1000$	
3	40	$n \leq 100\,000$	
4	10	–	

## Задача G. Обмены в перестановке

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

Дана перестановка  $a$  чисел от 1 до  $n$ , а также набор из  $m$  пар индексов. За один ход разрешается выбрать одну из этих  $m$  пар и поменять элементы на соответствующих позициях местами (перестановка, соответственно, изменится). Вы можете сделать произвольное количество ходов (в частности, разрешается не делать ни одного хода).

Определим возрастающую подпоследовательность длины  $k$  как набор индексов  $j_1, j_2, \dots, j_k$ , для которых выполняются два условия:

- $1 \leq j_1 < j_2 < \dots < j_k \leq n$ ;
- $a_{j_1} < a_{j_2} < \dots < a_{j_k}$ .

Какой максимально возможной длины наибольшей возрастающей подпоследовательности можно достичь при правильных обменах элементов?

### Формат входных данных

В первой строке заданы два числа  $n$  и  $m$  ( $1 \leq n \leq 10^4, 0 \leq m \leq \min(10^5, \frac{n \cdot (n-1)}{2})$ ) — длина перестановки и количество пар позиций, которые можно обменивать между собой.

В следующей строке через пробел заданы  $n$  различных целых чисел  $a_i$  ( $1 \leq a_i \leq n$ ) — элементы перестановки.

Каждая из следующих  $m$  строк содержит по два числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ ) — индексы позиций, элементы на которых можно менять. Гарантируется, что ни одна пара не встречается дважды.

### Формат выходных данных

Выведите одно целое число — максимально возможную длину наибольшей возрастающей подпоследовательности перестановки после обменов.

### Примеры

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

### Замечание

Рассмотрим перестановку из первого примера.

[5, 2, 4, 6, 3, 1]

Поменяем местами элементы на позициях 5 и 6.

[5, 2, 4, 6, 1, 3].

Теперь поменяем местами элементы на позициях 1 и 5.

[1, 2, 4, 6, 5, 3].

Длина наибольшей возрастающей подпоследовательности в такой перестановке равняется 4.

Соответствующая подпоследовательность: [1, 2, 4, 6, 5, 3].

## Система оценки

Тесты к этой задаче состоят из шести подзадач. Баллы за каждую подзадачу ставятся только при прохождении всех тестов подзадачи и всех тестов **необходимых** подзадач. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$m$		
0	0	–	–	–	Тесты из условия
1	7	$n \leq 8$	$m = 0$	–	
2	8	$n \leq 8$	–	0, 1	–
3	10	$n \leq 20$	–	0, 1, 2	–
4	17	$n \leq 100$	–	0, 1, 2, 3	–
5	18	$n \leq 1000$	–	0, 1, 2, 3, 4	–
6	40	–	–	0, 1, 2, 3, 4, 5	<b>Offline-проверка</b>