

## Задача А. Конфеты

Имя входного файла: `candies.in`  
Имя выходного файла: `candies.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Галантерейщик Бонасье очень любит конфеты. Однажды за ужином его жена Констанция выложила в ряд  $n$  конфет. У каждой конфеты есть тип  $p_i$ . Констанция предложила мужу выбрать последовательность подряд идущих конфет с одним условием — в этой последовательности должно быть ровно два различных типа конфет. Бонасье просит вас узнать, какое максимальное количество конфет он может взять, учитывая условие Констанции.

### Формат входного файла

В первой строке находится одно натуральное число  $n$  ( $n \leq 10^6$ ) — количество конфет. Во второй строке содержатся  $n$  целых чисел  $p_i$  ( $1 \leq p_i \leq 10^9$ ), где  $p_i$  — тип  $i$ -й конфеты.

### Формат выходного файла

Выведите максимальное количество конфет, которые может взять Бонасье.

### Пример

<code>candies.in</code>	<code>candies.out</code>
6 3 3 1 2 2 1	4
2 1 1	0

### Комментарий

В первом примере на столе лежит три типа конфет 1, 2 и 3. Бонасье может взять первые три конфеты с типами 3, 3, 1, а может взять последние четыре конфеты 1, 2, 2, 1. Значит, максимальное количество конфет, которое он может взять равно четырем.

Во втором примере существует всего один тип конфет, значит Бонасье не сможет взять ни одной конфеты.

### Система оценивания

Первая группа тестов состоит из тестов, для которых выполняются ограничения  $n \leq 100$ . Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 30 баллов.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение  $n \leq 1000$ . Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 30 баллов.

Третья группа тестов состоит из тестов, для которых выполняется ограничение  $n \leq 10^6$ . Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 40 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».

## Задача В. Осада Ла-Рошели

Имя входного файла:	<code>stairs.in</code>
Имя выходного файла:	<code>stairs.out</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Во время осады Ла-Рошели Д'Артаньян решил продумать каждую мелочь. Его интересует коммуникация крепости.

Д'Артаньяну известно, что пушки в Ла-Рошели расположены на двух этажах. При этом на каждом этаже есть  $n$  комнат, в каждой из которых есть пушка, а также лестница на другой этаж. Каждая комната соединена дверными проемами с двумя соседними на этаже, таким образом что если все время двигаться в одном направлении, можно по кругу обойти весь этаж. Для удобства Д'Артаньян занумеровал комнаты по порядку таким образом, что комната 1 граничит с комнатами  $n$  и 2, комната 2 — с 1 и 3, и так далее. При этом комнаты, соединенные лестницей, имеют одинаковый номер.

Гугеноты используют связных для коммуникации между стрелками, которые перемещается между ними. Длиной пути связного при проходе между пушками Д'Артаньян считает количество дверных проемов (при переходе с одного этажа на другой дверных проемов нет), которое нужно пройти, чтобы попасть от одной пушки к другой. Связные идеально знают крепость и всегда ходят по кратчайшему из возможных маршрутов.

Д'Артаньян хочет предложить кардиналу обрушить некоторые лестницы, чтобы увеличить длину пути связных и ослабить гугенотов. Для того, чтобы убедиться в том, что план действенный, Д'Артаньян хочет посчитать, какие будут длины путей связных при разных вариантах развития событий. Его интересуют только те маршруты связных, которые начинаются на одном этаже, а заканчиваются на другом.

Во времена мушкетеров не было компьютеров, зато вы сейчас можете попробовать решить эту задачу.

### Формат входного файла

В первой строке находится два натуральных числа  $n, m$  ( $3 \leq n \leq 10^5$ ,  $1 \leq m \leq 2 \cdot 10^5$ ) — количество пушек на этаже и количество событий.

В каждой из следующих  $m$  строк заданы события в хронологическом порядке, в следующем формате: сначала идет натуральное число  $t$  ( $1 \leq t \leq 2$ ) — тип запроса,

- если  $t = 1$ , то далее следует одно натуральное число  $x$  ( $1 \leq x \leq n$ ) — номер лестницы, которая будет разрушена;
- если  $t = 2$ , то далее следуют два натуральных числа  $x, y$  ( $1 \leq x, y \leq n$ ) — номера комнат, на разных этажах.

Гарантируется, что всегда будет хотя бы одна неразрушенная лестница.

### Формат выходного файла

Для каждого запроса второго типа выведите длину кратчайшего пути между двумя комнатами.

## Пример

stairs.in	stairs.out
4 4	1
2 1 4	1
2 4 1	2
2 2 4	0
2 3 3	
6 5	4
1 1	2
1 2	
1 3	
2 1 3	
2 1 1	

## Комментарий

В первом примере лестницы не разрушаются, поэтому связанные могут всегда переходить на другой этаж в стартовой комнате.

Во втором примере перейти из 1-ой комнаты одного этажа в 3-ю комнату другого можно, например, перейдя на другой этаж в 6-ой комнате, лестница в которой будет цела. Аналогично можно добраться и из 1-ой комнаты в 1-ю.

## Система оценивания

Первая группа тестов состоит из тестов, для которых выполняется ограничение  $n, m \leq 100$  и нет запросов первого типа. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 23 балла.

Вторая группа тестов состоит из тестов, для которых выполняется ограничение  $n \leq 5000$ ,  $m \leq 10000$ . Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 24 балла.

Третья группа тестов состоит из тестов, для которых выполняются полные ограничения. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 53 балла.

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».

## Задача С. Парады Ришелье

Имя входного файла: `parades.in`  
Имя выходного файла: `parades.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Сегодня кардинал Ришелье решил построить своих гвардейцев. На плацу построено  $n$  шеренг. Каждый гвардеец в шеренге одет в форму некоторого цвета. Всего существует 26 цветов формы. Каждому типу формы соответствует своя буква латинского алфавита.

Можно считать, что каждая шеренга задается строкой  $s_i$ , в начале строки находится начало шеренги, в конце строки — конец шеренги.

Планируется провести несколько показательных парадов. У кардинала Ришелье есть строка  $t$ , с помощью которой он будет составлять каждый из парадов. Всего будет проведено  $k$  парадов следующим образом: для  $j$ -го парада Ришелье вычеркнет из строки  $t$  некоторую подстроку длиной  $j$ , оставшуюся строку он обозначит  $t_j$ , затем выберет все шеренги, префикс которых равен  $t_j$ , и они будут задействованы в  $j$ -м параде.

От вас требуется для каждого  $j$  от 1 до  $k$  найти, какое максимальное количество шеренг можно задействовать в  $j$ -м параде.

### Формат входного файла

В первой строке находится единственная строка  $t$  ( $1 \leq |t| \leq 10^5$ ), состоящая из строчных латинских букв.

В следующей строке находятся два натуральных числа  $n, k$  ( $1 \leq n \leq 10^5, 1 \leq k < |t|$ ) — количество построенных на плацу шеренг и количество парадов соответственно.

В каждой из следующих  $n$  строк находится описание шеренги  $s_i$  — непустая строка, состоящая из строчных латинских букв.

Суммарная длина  $s_i$  не превосходит  $10^5$ .

### Формат выходного файла

В единственной строке выведите  $k$  чисел — максимальное количество шеренг, участвующих в  $j$ -м параде.

### Пример

<code>parades.in</code>	<code>parades.out</code>
abcdef 3 3 abcefx abcdfab abcdef	2 2 3
abcab 3 4 aaba aabz ad	0 2 0 3

### Система оценивания

Первая группа тестов состоит из тестов, для которых выполняются ограничения  $n, |t| \leq 300$ . Суммарная длина  $s_i$  не превосходит 300. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 21 балл.

Вторая группа тестов состоит из тестов, для которых выполняются ограничения  $n, |t| \leq 2000$ . Суммарная длина  $s_i$  не превосходит 2000. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 28 баллов.

Третья группа тестов состоит из тестов, для которых выполняются полные ограничения. Баллы за эту группу начисляются только при прохождении всех тестов группы. Стоимость группы составляет 51 балл.

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».

## Задача D. Фальшивая монета

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Это интерактивная задача.

Д'Артаньян решил подарить Констанции на День Святого Валентина замечательное кольцо с бриллиантами. Посчитав все свои сбережения, он выяснил, что у него есть ровно  $2a + 1$  монета, половина из которых золотые, а половина — серебряные (то есть у него либо  $a$  золотых и  $a + 1$  серебряная монета, либо наоборот). Однако Д'Артаньян точно помнил, что у него было четное число монет, то есть видимо кто-то подложил ему еще одну монету, которая наверняка фальшивая. Д'Артаньян знает, что настоящие серебрянные и золотые монеты весят одинаково, а фальшивая может весить по-разному: если она золотая, то она весит меньше настоящей, а если серебряная, то больше.

К счастью, у мушкетера нашлись чашечные весы, с помощью которых он может брать две кучки монет, класть их на левую и правую чаши весов и узнавать, какая из кучек монет тяжелее (или что они весят одинаково). Времени у Д'Артаньяна мало, поэтому он успеет провести только  $t$  взвешиваний. Помогите ему отыскать фальшивую монету.

### Протокол взаимодействия с программой жюри:

В самом начале программа жюри сообщает вашей программе три натуральных числа  $a, b, t$  ( $|a - b| = 1$ ) — количество золотых и серебряных монет соответственно и количество ходов, за которое надо найти фальшивую монету. Золотые монета имеют номера от 1 до  $a$ , серебрянные - от  $a + 1$  до  $a + b$ .

Во время взаимодействия вашей программы с программой жюри несколько раз повторяются следующие действия:

- Если вы уверены, что знаете номер фальшивой монеты, в единственной строке выведите  $0 x$ , где  $x$  — номер фальшивой монеты, и завершите программу;
- Если нет, ваша программа сообщает программе жюри две последовательности чисел в двух разных строках. В первой строке записаны номера монет, которые следует положить на левую чашу весов в формате  $k a_1 a_2 \dots a_k$ . Во второй строке записаны номера монет, которые следует положить на правую чашу весов в том же формате.
- программа жюри сообщает вашей программе:
  - «<», если монеты на левой чаше имеют вес меньше монет на правой;
  - «=», если вес чаш одинаковый;
  - «>», если монеты на левой чаше имеют вес больше монет на правой
- После этого действия выше выполняются заново

### Пример

<code>stdin</code>	<code>stdout</code>
1 2 2 =	1 1 1 3 0 2
2 1 2 <	1 1 1 2 0 1

## Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output)`;
- В C/C++: `fflush(stdout)`;
- В Java: `System.out.flush()`;
- В Python: `sys.stdout.flush()`;

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

## Система оценивания

Каждый тест в этой задаче оценивается независимо и стоит 10 баллов.

- Первые два теста — тесты из условия, они оцениваются в 0 баллов.
- В третьем тесте  $a = 13, b = 14, t = 5$ ;
- В четвертом тесте  $a = 14, b = 13, t = 5$ ;
- В пятом тесте  $a = 13, b = 14, t = 4$ ;
- В шестом тесте  $a = 14, b = 13, t = 4$ ;
- В седьмом тесте  $a = 29524, b = 29525, t = 20$ ;
- В восьмом тесте  $a = 29525, b = 29524, t = 20$ ;
- В девятом тесте  $a = 29524, b = 29525, t = 12$ ;
- В десятом тесте  $a = 29525, b = 29524, t = 12$ ;
- В одиннадцатом тесте  $a = 29524, b = 29525, t = 11$ ;
- В двенадцатом тесте  $a = 29525, b = 29524, t = 11$ ;

Обратите внимание на возможность узнать результат проверки вашего решения на всех тестах, нажав на ссылку «Request feedback» на вкладке «Runs».