



[ЗАДАЧИ](#)
[ОТОСЛАТЬ](#)
[СТАТУС](#)
[ПОЛОЖЕНИЕ](#)
[ЗАПУСК](#)

В. Бал в БерлГУ

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

По случаю 100500-летия Берляндского государственного университета совсем скоро состоится бал! Уже n юношей и m девушек во всю репетируют вальс, менуэт, полонез и кадрили.

Известно, что на бал будут приглашены несколько пар юноша-девушка, причем уровень умений танцевать партнеров в каждой паре должен отличаться не более чем на единицу.

Для каждого юноши известен уровень его умения танцевать. Аналогично, для каждой девушки известен уровень ее умения танцевать. Напишите программу, которая определит наибольшее количество пар, которое можно образовать из n юношей и m девушек.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 100$) — количество юношей. Вторая строка содержит последовательность a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), где a_i — умение танцевать i -го юноши.

Аналогично, третья строка содержит целое m ($1 \leq m \leq 100$) — количество девушек. В четвертой строке содержится последовательность b_1, b_2, \dots, b_m ($1 \leq b_j \leq 100$), где b_j — умение танцевать j -й девушки.

Выходные данные

Выведите единственное число — искомое максимальное возможное количество пар.

Примеры

входные данные	Скопировать
4 1 4 6 2 5 5 1 5 7 9	
выходные данные	Скопировать
3	

входные данные	Скопировать
4 1 2 3 4 4 10 11 12 13	
выходные данные	Скопировать
0	

входные данные	Скопировать
5 1 1 1 1 1 3 1 2 3	
выходные данные	Скопировать
2	



С. Произведение трех чисел

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам задано целое число n . Найдите три **различных целых числа** a, b, c таких, что $2 \leq a, b, c$ и $a \cdot b \cdot c = n$ или скажите, что невозможно сделать это.

Если существует несколько подходящих ответов, вы можете вывести любой.

Вам нужно ответить на t независимых наборов входных данных.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

Следующие n строк описывают наборы входных данных. i -й набор входных данных задан с новой строки в виде одного целого числа n ($2 \leq n \leq 10^9$).

Выходные данные

Выведите ответ для каждого набора входных данных. Выведите «NO» если невозможно представить n как $a \cdot b \cdot c$ для каких-то **различных целых чисел** a, b, c таких, что $2 \leq a, b, c$.

Иначе выведите «YES» и **любое** такое возможное представление.

Пример

входные данные	Скопировать
5 64 32 97 2 12345	
выходные данные	Скопировать
YES 2 4 8 NO NO NO YES 3 5 823	



В. Случайные команды

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Для участия в соревнованиях n участников были разбиты некоторым образом на m команд так, чтобы в каждой команде был хотя бы один участник. После соревнований каждая пара участников из одной команды стала друзьями.

Ваша задача — написать программу, которая определит, какое минимальное и какое максимальное количество пар друзей могло образоваться после соревнования.

Входные данные

В единственной строке содержатся два целых числа n и m , разделенных одним пробелом ($1 \leq m \leq n \leq 10^9$) — количество участников и количество команд соответственно.

Выходные данные

Требуется вывести два целых числа k_{min} и k_{max} — минимальное возможное количество пар друзей и максимальное возможное количество пар друзей соответственно.

Примеры

входные данные	Скопировать
5 1	
выходные данные	Скопировать
10 10	
входные данные	Скопировать
3 2	
выходные данные	Скопировать
1 1	
входные данные	Скопировать
6 3	
выходные данные	Скопировать
3 6	

Примечание

В первом примере все участники попадают в одну команду, поэтому в любом случае будет образовано ровно десять пар друзей.

Во втором примере при любом разбиении в одной из команд будет два участника, а в другой — один. В таком случае количество пар друзей всегда будет равно одному.

В третьем примере минимальное количество друзей выходит при разбиении на команды по 2 человека, а максимальное — при разбиении на команды размерами 1, 1 и 4 человек.



Е. Упорядочивание овец

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вы играете в игру «Упорядочивание овец». Цель этой игры — сделать так, чтобы овцы выстроились в ряд. Уровень в игре описывается строкой длины n , состоящей из символов `'.'` (пустое пространство) и `'*'` (овечка). За один ход вы можете передвинуть любую овечку на одну клетку влево или на одну клетку вправо, если соответствующая клетка **существует и пуста**. Игра заканчивается, как только овцы выстроились в ряд, то есть между любыми овечками не должно быть пустых клеток.

Например, если $n = 6$ и уровень описывается строкой `«**.*..»`, тогда возможен следующий сценарий игры:

- овечка на позиции 4 двигается вправо, состояние уровня: `«**.*..»`;
- овечка на позиции 2 двигается вправо, состояние уровня: `«*. *.*..»`;
- овечка на позиции 1 двигается вправо, состояние уровня: `«. *.*.*..»`;
- овечка на позиции 3 двигается вправо, состояние уровня: `«. *.*.*.*..»`;
- овечка на позиции 2 двигается вправо, состояние уровня: `«. . * * * . »`;
- овцы выстроились в ряд, игра завершается.

Для заданного уровня определите минимальное количество ходов, которое нужно сделать, чтобы пройти уровень.

Входные данные

В первой строке содержится одно целое число t ($1 \leq t \leq 10^4$). Далее следуют t наборов входных данных.

В первой строке каждого набора входных данных содержится одно целое число n ($1 \leq n \leq 10^6$).

Во второй строке каждого набора входных данных содержится строка длины n , состоящая из символов `'.'` (пустое пространство) и `'*'` (овечка) — описание уровня.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^6 .

Выходные данные

Для каждого набора входных данных выведите одно число — минимальное количество ходов, которое нужно сделать, чтобы пройти уровень.

Пример

входные данные	Скопировать
<pre>5 6 **.*.. 5 ***** 3 .*. 3 ... 10 *.*...*.*</pre>	
выходные данные	Скопировать
<pre>1 0 0 0 9</pre>	

D. Эпическая трансформация

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам задан массив a длины n , состоящий из целых чисел. Вы можете применять следующую операцию, состоящую из нескольких шагов, над массивом a **ноль** или более раз:

- вы выбираете два **различных** числа в массиве a_i и a_j ;
- вы удаляете i -й и j -й элементы из массива.

Например, если $n = 6$ и $a = [1, 6, 1, 1, 4, 4]$, то вы можете произвести следующую последовательность операций:

- выбираем $i = 1, j = 5$. Массив a становится равным $[6, 1, 1, 4]$;
- выбираем $i = 1, j = 2$. Массив a становится равным $[1, 4]$.

Каким может быть минимальный размер массива после применения к нему некоторой последовательности операций?

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следуют t наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина массива a .

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальный возможный размер массива после применения к нему некоторой последовательности операций.

Пример

входные данные	Скопировать
5 6 1 6 1 1 4 4 2 1 2 2 1 1 5 4 5 4 5 4 6 2 3 2 1 3 1	
выходные данные	Скопировать
0 0 2 1 0	



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Увеличения в группах

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Дан массив a длины n . Чтобы вычислить штраф, нужно выполнить следующие действия:

1. Разбить массив a на две (возможно, пустых) подпоследовательности $^\dagger s$ и t такие, что каждый элемент a находится либо в s , либо в t .
2. Для массива b длины m определим штраф $p(b)$ массива b как количество индексов i между 1 и $m - 1$ таких, что $b_i < b_{i+1}$.
3. Общий штраф, который вы получите, равен $p(s) + p(t)$.

Найдите минимально возможный штраф, который вы получите, если выполните вышеописанные действия оптимально.

† Последовательность x является подпоследовательностью y , если x может быть получена из y удалением нескольких (возможно, ни одного или всех) элементов.

‡ Некоторыми допустимыми способами разбиения массива $a = [3, 1, 4, 1, 5]$ на (s, t) являются $([3, 4, 1, 5], [1])$, $([1, 1], [3, 4, 5])$ и $([], [3, 1, 4, 1, 5])$, в то время как некоторыми недопустимыми способами разбиения a являются $([3, 4, 5], [1])$, $([3, 1, 4, 1], [1, 5])$ и $([1, 3, 4], [5, 1])$.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длину массива a .

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — элементы массива a .

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число, представляющее собой минимальный возможный штраф, который вы можете получить.

Пример

входные данные	Скопировать
5 5 1 2 3 4 5 8 8 2 3 1 1 7 4 3 5 3 3 3 3 3 1 1 2 2 1	
выходные данные	Скопировать
3 1 0 0 0	

Примечание

В первом наборе входных данных возможный способ разбиения a — $s = [2, 4, 5]$ и $t = [1, 3]$. Штраф равняется $p(s) + p(t) = 2 + 1 = 3$.

Во втором наборе входных данных возможный способ разбиения a — $s = [8, 3, 1]$ и $t = [2, 1, 7, 4, 3]$. Штраф равняется $p(s) + p(t) = 0 + 1 = 1$.

В третьем наборе входных данных возможным способом разбиения a является $s = []$ и $t = [3, 3, 3, 3, 3]$. Штраф равняется $p(s) + p(t) = 0 + 0 = 0$.

В. Правильная скобочная подпоследовательность

ограничение по времени на тест: 5 seconds
ограничение по памяти на тест: 256 megabytes

Напомним, что скобочная последовательность называется правильной, если путем вставки в нее символов «+» и «1» можно получить из нее корректное математическое выражение. Например, последовательности «`(())()`», «`()`» и «`(()())`» — правильные, в то время как «`)`», «`(()`» и «`(())()`» — нет.

Однажды Васе попалась скобочная последовательность. Он решил удалить из нее некоторые скобки так, чтобы последовательность стала правильной. Какую наибольшую длину может иметь получившаяся правильная скобочная последовательность?

Входные данные

В первой строке входного файла записана непустая строка, состоящая из символов «`(`» и «`)`». Её длина не превосходит 10^6 .

Выходные данные

Выведите длину наибольшей правильной скобочной подпоследовательности.

Примеры

входные данные	Скопировать
<code>(())(</code>	
выходные данные	Скопировать
<code>4</code>	
входные данные	Скопировать
<code>(())()</code>	
выходные данные	Скопировать
<code>6</code>	

