

Partsort

СЕЗОН 10 – ВТОРИ РУНД



Дадена е редица, която е пермутация на числата от 1 до N . Казваме, че редицата е сортирана, ако за всяко $i = 1, 2, \dots, N - 1$ е изпълнено неравенството $A_i < A_{i+1}$, където с A_i означаваме i -тото число в редицата. В недалечното минало компютрите са разполагали с доста по-малко оперативна памет и именно заради това сортирането на редици не е било толкова тривиална задача.

Нека знаем, че в паметта може да поддържаеме едновременно K от елементите на редицата. Ето защо сортираме числата с индекси между 1 и K , след това между 2 и $K + 1$ и т.н. до тези между $N - K + 1$ и N . За съжаление, това невинаги е достатъчно, за да сортираме редицата. Например, ако $N = 5$, $K = 3$ и $A = \{4, 5, 3, 1, 2\}$, последователно получаваме следните промени: $\{3, 4, 5, 1, 2\} \rightarrow \{3, 1, 4, 5, 2\} \rightarrow \{3, 1, 2, 4, 5\}$. Така за една стъпка от началната редица получаваме $\{3, 1, 2, 4, 5\}$ и се налага да направим втора стъпка, за да сортираме редицата.

Напишете програма, която намира броя на стъпките, които ще извърши алгоритъмът, за да сортира редицата. Програмата трябва да обработва T тестови случая.

Вход

От първия ред на входния файл `partsort.in` се въвежда числото T . Следват T реда, всеки от които описва по един тестов случай във формат – $N, K, A_1, A_2, \dots, A_N$.

Изход

На N реда от изходния файл `partsort.out` изведете по едно число, равно на търсения брой повторения на описания алгоритъм, за да се сортира редицата от съответния тестов случай.

Ограничения

$$2 \leq K \leq N \leq 10\,000$$

Сумата от $N \div K$ за всички тестови случаи е не по-голяма от 100.

Пример

Вход	Изход
3	2
5 3 4 5 3 1 2	0
3 2 1 2 3	1
7 7 7 6 5 4 3 2 1	