

第三題：窮根尋葉 (Leaves)

問題敘述

在二元搜尋樹 (binary search tree) 中，每一個節點有一個鍵值 (key)，比根節點鍵值小的節點會被放在根節點的左子樹，比根節點鍵值大的節點會被放在根節點的右子樹，並且左、右子樹亦滿足上述大小條件。

如圖 1，根節點的鍵值為 5，因此比 5 小的節點要放在左子樹中，比 5 大的節點要放在右子樹中；同理，以 7 為根節點的子樹中，節點 6 要放在節點 7 的左子樹，節點 8 要放在節點 7 的右子樹。當我們使用前序法 (preorder) 列舉出某二元搜尋樹上所有節點的鍵值時，實際上只要利用這個前序序列，就可以還原出原本的二元搜尋樹。比如說，圖 1 中的二元搜尋樹，其所對應的前序序列為 5, 3, 1, 7, 6, 8；我們只要將這個序列的數字，依序插入一開始是空的二元搜尋樹中，就可以還原出如圖 1 的二元搜尋樹。

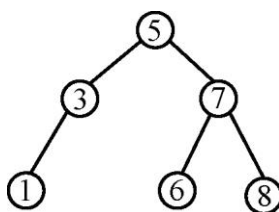


圖 1. 二元搜尋樹

二元搜尋樹中，沒有左、右子樹的節點，稱為葉節點 (leaf node)。圖 1 中共有 3 個葉節點，分別是節點 1、6、8。請你撰寫一個程式，讀入一個包含 n 個相異整數的序列以及一個整數 k ，找出有多少連續的子序列是一棵恰好有 k 個葉節點的二元搜尋樹的前序序列。

輸入格式

輸入第一列包含兩個以空白隔開的正整數 n, k 。第二列包含 n 個相異整數。

輸出格式

輸出單一整數，代表有多少個連續子序列滿足題目要求。

輸入範例 1 5 1 1 2 3 4 5	輸出範例 1 15
輸入範例 2 5 2 1 2 3 4 5	輸出範例 2 0

輸入範例 3 5 2 2 1 4 3 5	輸出範例 3 4
-----------------------------------	--------------------

評分說明

本題共有 5 個子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	17	$1 \leq k \leq n \leq 100$ 且輸入序列為 1 到 n 的排列。
2	8	$k = 1$ 且 $1 \leq n \leq 100,000$ 且所有數字絕對值不超過 10^9 。
3	26	$1 \leq k \leq n \leq 3,000$ 且輸入序列為 1 到 n 的排列。
4	22	$1 \leq k \leq n \leq 20,000$ 且輸入序列為 1 到 n 的排列。
5	27	$1 \leq k \leq n \leq 500,000$ 且所有數字絕對值不超過 10^9 。