Elections are in progress!

Given an array of the numbers of votes given to each of the candidates so far, and an integer kequal to the number of voters who haven't cast their vote yet, find the number of candidates who still have a chance to win the election.

The winner of the election must secure strictly more votes than any other candidate. If two or more candidates receive the same(maximum) number of votes, assume there is no winner at all.

**Example**

For votes = [2, 3, 5, 2] and k = 3, the output should be  
electionsWinners(votes, k) = 2.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer votes**

A non-empty array of non-negative integers. Its *ith* element denotes the number of votes cast for the *ith* candidate.

*Constraints:*  
4 ≤ votes.length ≤ 105,  
0 ≤ votes[i] ≤ 104.

* **[input] integer k**

The number of voters who haven't cast their vote yet.

*Constraints:*  
0 ≤ k ≤ 105.

* **[output] integer**

<https://codefights.com/arcade/code-arcade/well-of-integration/8RiRRM3yvbuAd3MNg>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int electionsWinners(int[] votes, int k)

{

int max = votes.Max();

int contMax = votes.Count(f => f == max);

// Console.WriteLine(contMax);

int ans = 0;

for (int i = 0; i < votes.Length; i++)

{

if (votes[i] == max && contMax == 1)

{

ans++;

}

else if (votes[i] + k > max)

{

ans++;

}

}

return ans;

}

static void Main(string[] args)

{

int[] votes = { 2, 3, 5, 2 };

int k = 3;

//int[] votes = {5, 1, 3, 4, 1,5,5,5,5,5};

//int[] votes = { 5, 1, 3, 4, 1};

//int k = 0;

Console.WriteLine( electionsWinners(votes, k));

Console.ReadLine();

}

}

}