**Java - Competitive Programming Challenge**

**Question 10**

**Question Description:**

Given an integer k and a list of integers, count the number of distinct valid pairs of integers (a, b) in the list for which a + k = b. Two pairs of integers (a, b) and (c, d) are considered distinct if at least one element of (a, b) does not also belong to (c, d).

Example:

n = 4

numbers = [1, 1, 1, 2]

k = 1

This array has two different valid pairs: (1, 1) and (1, 2). For k = 1, there is only 1 valid pair which satisfies a + k = b: the pair (a, b) = (1, 2).

**Function Description:**

Complete the function countPairs in the editor below.

countPairs has the following parameter(s):

* int numbers[n]: array of integers
* int k: target difference

**Returns:**

* int: number of valid (a, b) pairs in the numbers array that have a difference of k

**Constraints:**

* 2 ≤ n ≤ 2 × 105
* 0 ≤ numbers[i] ≤ 109, where 0 ≤ i < n
* 0 ≤ k ≤ 109

**Question 11**

**Question Description:**

A climber is trying to reach a flag that is some height above the ground. In the attempt to reach the flag, the climber can make any number of jumps up the rock wall where the flag is mounted. Movements can only be made up the wall, and the climber must end at exactly the height of the flag.

There are 2 types of jumps:

1. A jump of height 1.
2. A jump of height bigJump.

Determine the minimum number of jumps it will take the climber to reach the flag's exact height.

**Example:**

flagHeight = 8

bigJump = 3

The climber starts at height 0, takes two jumps of height bigJump and two of height 1 to reach exactly 8 units in 4 jumps.

**Function Description:**

Complete the function jumps in the editor below.

jumps has the following parameter(s):

* int flagHeight: an integer, the flag height
* int bigJump: an integer, the height of the second type of jump

**Returns:**

int: an integer, the minimum number of jumps necessary

Constraints:

1 ≤ bigJump, ﬂagHeight ≤ 109