Your best friend Bob asked you to count the number of integer pairs in a given array whose difference is equal to an input parameter \( k \). (PS: I hope Bob is still your friend by the end of this lab 😊). Assume that the elements in the array are unique. There are two ways to approach the problem:

1. Use sorting (\texttt{solve\_with\_Sort(int[] array, int k)}):
   a. Initialize a counter variable to 0: \( \text{counter} = 0 \)
   b. Sort the array using the Java library \( \text{Arrays.sort(array)}; \)
   c. Take two pointers, \textit{left} and \textit{right}, initially having values: \( \text{left} = 0, \text{right} = 0 \)
   d. While \( \text{right} < \text{arr.length} \):
      - Get the difference: \( \text{diff} = \text{array[right]} - \text{array[left]} \)
      - If \( \text{diff} == k \), increment \text{counter}, increment \text{left} and increment \text{right}.
      - If \( \text{diff} > k \), increment \text{left} pointer
      - If \( \text{diff} < k \), increment \text{right} pointer
   e. \text{return \text{counter}}

2. Use hashing (\texttt{solve\_with\_Hash(int[] array, int k)})
   a. Initialize a counter variable to 0: \( \text{counter} = 0 \)
   b. Insert all elements of \textit{array} in a hashtable
   c. For every element in \textit{array}:
      - Look for \( \text{array[i]} + k \) in the hash map, if found then increment \text{counter}.
      - Look for \( \text{array[i]} - k \) in the hash map, if found then increment \text{counter}.
      - Remove \text{array[i]} from hash table.
   d. \text{return \text{counter}}

For example:

\textbf{Input:} \text{array[]} = \{1, 5, 3, 4, 2\}, \text{k} = 3

\textbf{Output:} 2

\textbf{Explanation:} There are 2 pairs with difference 3, the pairs are \{4, 1\} and \{5, 2\}
import java.util.Arrays;
import java.util.HashSet;

public class Lab9 {

    public static void main(String []args){

        int[] arr = {1, 5, 3, 4, 2};
        int k = 3;

        long start = System.currentTimeMillis();
        System.out.println("Count is "+ solve_with_Hash(arr,k));  // should be 2
        System.out.println("Total execustion time (taken by solve_with_Hash):
                           +"+(System.currentTimeMillis()-start));

        start = System.currentTimeMillis();
        System.out.println("\nCount is "+ solve_with_Sort(arr,k));  // should be 2
        System.out.println("Total execustion time (taken by solve_with_Sort):
                           +"+(System.currentTimeMillis()-start));

    }

    public static int solve_with_Sort(int[] arr, int k){
        int counter = 0;

        // INSERT CODE HERE

        return counter;
    }

    public static int solve_with_Hash(int[] arr, int k){
        int counter = 0;
        HashSet<Integer> hash = new HashSet<Integer>();

        // INSERT CODE HERE

        return counter;
    }
}