Java

```
class FindSumPairs {

    public FindSumPairs(int[] nums1, int[] nums2) {

    }

    public void add(int index, int val) {

    }

    public int count(int tot) {

    }
}

/**
 * Your FindSumPairs object will be instantiated and called as such:
 * FindSumPairs obj = new FindSumPairs(nums1, nums2);
 * obj.add(index,val);
 * int param_2 = obj.count(tot);
 */
--------------------------------------------------------------------
```

JavaScript

```
/**
 * @param {number[]} nums1
 * @param {number[]} nums2
 */
var FindSumPairs = function(nums1, nums2) {
};

/**
 * @param {number} index
 * @param {number} val
 * @return {void}
 */
FindSumPairs.prototype.add = function(index, val) {

};

/**
 * @param {number} tot
 * @return {number}
 */
FindSumPairs.prototype.count = function(tot) {

};

/**
 * Your FindSumPairs object will be instantiated and called as such:
 * var obj = new FindSumPairs(nums1, nums2)
 * obj.add(index,val)
 * var param_2 = obj.count(tot)
 */
--------------------------------------------------------------------
```

TypeScript

```
class FindSumPairs {
    constructor(nums1: number[], nums2: number[]) {
    }

    add(index: number, val: number): void {

    }

    count(tot: number): number {

    }
}

/**
 * Your FindSumPairs object will be instantiated and called as such:
 * var obj = new FindSumPairs(nums1, nums2)
 * obj.add(index,val)
 * var param_2 = obj.count(tot)
 */
--------------------------------------------------------------------
```

C++

```
class FindSumPairs {
public:
    FindSumPairs(vector<int>& nums1, vector<int>& nums2) {

    }

    void add(int index, int val) {

    }
    int count(int tot) {

    }
};

/**
 * Your FindSumPairs object will be instantiated and called as such:
 * FindSumPairs* obj = new FindSumPairs(nums1, nums2);
 * obj->add(index,val);
 * int param_2 = obj->count(tot);
 */
--------------------------------------------------------------------
```

C#

```
public class FindSumPairs {

    public FindSumPairs(int[] nums1, int[] nums2) {

    }

    public void Add(int index, int val) {

    }

    public int Count(int tot) {

    }
}

/**
 * Your FindSumPairs object will be instantiated and called as such:
```

```
 * FindSumPairs obj = new FindSumPairs(nums1, nums2);
 * obj.Add(index,val);
 * int param_2 = obj.Count(tot);
 */
-----------------------------------------------------------------------
```

Kotlin

```
class FindSumPairs(nums1: IntArray, nums2: IntArray) {

    fun add(index: Int, `val`: Int) {

    }

    fun count(tot: Int): Int {

    }

}

/**
 * Your FindSumPairs object will be instantiated and called as such:
 * var obj = FindSumPairs(nums1, nums2)
 * obj.add(index,`val`)
 * var param_2 = obj.count(tot)
 */
--------------------------------------------------------------------
```

Go

```
type FindSumPairs struct {
}


func Constructor(nums1 []int, nums2 []int) FindSumPairs {

}


func (this *FindSumPairs) Add(index int, val int)  {

}


func (this *FindSumPairs) Count(tot int) int {

}


/**
 * Your FindSumPairs object will be instantiated and called as such:
 * obj := Constructor(nums1, nums2);
 * obj.Add(index,val);
 * param_2 := obj.Count(tot);
 */
--------------------------------------------------------------------
```