```
// Given an array of size N-1 such that it only contains
distinct integers in the range of 1 to N. Find the missing
element.
// Expected Time Complexity: O(N)
// Expected Auxiliary Space: O(1)
// 1 ≤ N ≤ 10^6 means O(nlogn)
```

The length of the array is **n**. So, the sum of all n+1 elements i.e. sum of numbers from **1** to **n+1** can be calculated using the formula ***n+1*(n+2)/2***. Now find the sum of all the elements in the array and subtract it from the sum of the first n+1 natural numbers, it will give us the value of the missing element.

Brute Force:

```
int MissingNumber(vector<int> &array, int n) // My approach: O(nlogn)
{
    sort(array.begin(), array.end()); // nlog(n)
    int count = 0;
    for (int i = 0; i < n; i++) // O(N)
    {
        count++;
        if (array[i] != count)
            return count;
    }
}
```

Optimized: Time complexity: O(n); Space complexity: O(1)

```
int sumTotal = (n+1)*(n+2)/2;
 for (int i = 0; i < n; i++)
    {
        sumTotal = sumTotal - array[i];
    }
    return sumTotal;
```