1.Easy\10.Missing_number.cpp

```cpp
/*
QUESTION:-
Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array.

Example 1:

Input: nums = [3,0,1]
Output: 2
Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.
Example 2:

Input: nums = [0,1]
Output: 2
Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.
*/

/*
APPROACH:-
-> Calculate the optimum sum i.e. sum when all elements were present
-> Calculate the actual array's sum
-> Return the optimum sum - actual sum
*/

// CODE:-
int missingNumber(vector<int> &nums)
{
    int n = nums.size();
    long long optimum_sum = (n * (n + 1)) / 2; // the sum if no number is absent
    long long actual_sum = 0;
    for (auto it : nums)
    {
        actual_sum += it;
    }
    return optimum_sum - actual_sum;
}

// TIME COMPLEXITY = O(N)
// SPACE COMPLEXITY = O(0)
```