1.Easy\11.Max_consecutive_1's.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:-
-> Traverse the entire array and within it run a loop while element's are equal to 1 and store the count
-> Update the ans as max(ans,cnt)
*/

// CODE:-
int findMaxConsecutiveOnes(vector<int> &nums)
{
    int ans = 0;
    for (int i = 0; i < nums.size(); i++)
    {
        int cnt = 0; // to store the count of consecutive 1's
        while (i < nums.size() && nums[i] == 1)
        {
            cnt++;
            i++;
        }
        ans = max(ans, cnt);
    }
    return ans;
}

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

```