



</> Problem

📖 Editorial

🕒 Submissions

💬 Comments

C++ (g++ 5.4)▾

Average Time: 30m

🕒 Start Timer



Missing And Repeating 🔒

Difficulty: Medium

Accuracy: 24.83%

Submissions: 498K+

Points: 4

Given an unsorted array **arr** of positive integers. One number '**A**' from set {1, 2,...,n} is missing and one number '**B**' occurs twice in array. Find numbers **A** and **B**.

Examples

Input: arr[] = [2, 2]**Output:** 2 1**Explanation:** Repeating number is 2 and smallest positive missing number is 1.**Input:** arr[] = [1, 3, 3]**Output:** 3 2**Explanation:** Repeating number is 3 and smallest positive missing number is 2.**Expected Time Complexity:** O(n)**Expected Auxiliary Space:** O(1)

Constraints:

 $2 \leq n \leq 10^5$ $1 \leq arr[i] \leq n$ [Try more examples](#)

Seen this question in a real interview before ?

[Yes](#)[No](#)

Company Tags

[Amazon](#)[Samsung](#)[D-E-Shaw](#)[Goldman Sachs](#)[MAQ Software](#)

Topic Tags

Related Interview Experiences

Related Articles

[🚩 Report An Issue](#)

If you are facing any issue on this page. Please let us know.

Discover your potential with Deutsche Bank. Innovate, grow, and succeed globally in your career [🔗](#)

```
1- //{ Driver Code Starts
2- #include <bits/stdc++.h>
3-
4- using namespace std;
5-
6-
7- // } Driver Code Ends
8-
9- class Solution {
10- public:
11-     vector<int> findTwoElement(vector<int>& arr) {
12-         int n = arr.size();
13-         vector<int> result(2); // result[0] = repeating, result[1] = missing
14-
15-         // Step 1: Use index marking to find the repeating number
16-         for (int i = 0; i < n; i++) {
17-             int absVal = abs(arr[i]);
18-             if (arr[absVal - 1] > 0) {
19-                 arr[absVal - 1] = -arr[absVal - 1]; // Mark the index as visited
20-             } else {
21-                 result[0] = absVal; // Repeating number found
22-             }
23-         }
24-
25-         // Step 2: Find the missing number
26-         for (int i = 0; i < n; i++) {
27-             if (arr[i] > 0) { // The index that was never visited (i.e., positive)
28-                 result[1] = i + 1; // Missing number found
29-                 break;
30-             }
31-         }
```

[Custom Input](#)[Compile & Run](#)[Submit](#)