**Form a number divisible by 3 using array digits: -**

**Easy Accuracy: 63.96% Submissions: 51K+ Points: 2**

You will be given an array **arr** of integers of length **N.**You can construct an integer from two integers by treating the integers as strings, and then concatenating them. For example, **19**and **4** can be used to construct **194**and **419.**

The task is to find whether it’s possible to construct an integer using all the digits of these numbers such that it would be **divisible by 3**.  
If it is possible then print **1** and if not print **0**.

**Example 1:**

**Input:** N = 3

arr = {40, 50, 90}

**Output:** 1

**Explanation:** One such number is 405090.

**Example 2:**

**Input:** N = 2

arr = {1, 4}

**Output:** 0

**Explanation:** The numbers we can form

are 14 and 41. But neither of them are

divisible by 3.

**Your Task:**  
You do not need to read input or print anything. Your task is to complete the function **isPossible()** which takes **N** and **arr** as input parameters and returns **1 i**f we can form a number by the digits of the given number, that would be divisible by 3, otherwise returns **0**.

**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 ≤ N, arr[i] ≤ 105

**Code: -**

//{ Driver Code Starts

// Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// User function Template for C++

class Solution {

public:

int isPossible(int N, int arr[]) {

// code here

int sum = 0;

for(int i = 0; i < N; ++i){

while(arr[i] > 0){

sum += (arr[i] % 10);

arr[i] /= 10;

}

}

return sum % 3 == 0;

}

};

//{ Driver Code Starts.

int main() {

int t;

cin >> t;

while (t--) {

int N;

cin >> N;

int arr[N];

for (int i = 0; i < N; i++) cin >> arr[i];

Solution ob;

cout << ob.isPossible(N, arr) << endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(N \* log10 N)**

**S.C: - O(1)**