**Find triplets with zero sum**

Submissions: [9635](https://practice.geeksforgeeks.org/problem_submissions.php?pid=700466)  Accuracy:

45.91%

   Difficulty: [Basic](https://practice.geeksforgeeks.org/Basic/1/0/)   Marks: 1

Show Topic Tags   

[Facebook](https://practice.geeksforgeeks.org/company/Facebook/)[Google](https://practice.geeksforgeeks.org/company/Google/)

Given an array **A** of **N** elements. The task is to complete the function which returns true if triplets exists in array A whose sum is zero else returns false.

**Input Format:**  
The first line of input contains an integer **T**, denoting the number of test cases. Then **T** test cases follow. The first line of each test case contains an integer N, denoting the number of elements in array. The second line of each test case contains N space separated values of the array.

**Output:**  
For each test case, output will be 1 if triplet exists else 0.

**Your Task:**  
Your task is to only complete the function to find triplets.

**Constrains:**  
1 <= T <= 100  
1 <= N <= 103  
-103 <= A <= 103

**Example:  
Input:**  
2  
5  
0 -1 2 -3 1  
3  
1 2 3

**Output:**  
1  
0

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/find-triplets-with-zero-sum/1/?ref=self#ExpectOP) option \*\*

<https://practice.geeksforgeeks.org/problems/find-triplets-with-zero-sum/1/?ref=self>

#include<bits/stdc++.h>

#include<stdlib.h>

#include<iostream>

#include <map>

using namespace std;

*/\*Please note that it's Function problem i.e.*

*you need to write your solution in the form of Function(s) only.*

*Driver Code to call/invoke your function is mentioned above.\*/*

*/\*You are required to complete the function below\*/*

bool findTriplets(int arr[], int n)

{

    map<int,int> m;

    //Your code here

    for(int i =0; i<n; i++) {

        m[arr[i]] = i;

    }

    for(int i =0; i<n-1; i++) {

        for(int j =i+1; j<n; j++) {

            if(m.find(-(arr[i]+arr[j])) != m.end()) {

                if(m[ -(arr[i]+arr[j])  ] != i && m[ -(arr[i]+arr[j])  ] != j) {

                    return true;

                }

            }

        }

    }

    return false;

}

int main()

{

*/\**

*int t;*

*cin>>t;*

*while(t--){*

*int n;*

*cin>>n;*

*int arr[n]={0};*

*for(int i=0;i<n;i++)*

*cin>>arr[i];*

*if(findTriplets(arr, n))*

*cout<<"1"<<endl;*

*else*

*cout<<"0"<<endl;*

*}*

*\*/*

    int arr[] = { 6 , 56, 93, -12, 26, 78, 79, 58, 53, 52, 51, 55, 77, -2, 61, -26, 91, 16, 100, -8, 72};

    int n = sizeof(arr)/sizeof(int);

    cout << findTriplets( arr,  n)  << endl;

    return 0;

}