Given an array **arr[]**of **n** integers. Check whether it contains a triplet that sums up to zero. 

**Note:**Return 1, if there is at least one triplet following the condition else return 0.

**Example 1:**

**Input**: n = 5, arr[] = {0, -1, 2, -3, 1}

**Output**: 1

**Explanation**: 0, -1 and 1 forms a triplet

with sum equal to 0.

**Example 2:**

**Input**: n = 3, arr[] = {1, 2, 3}

**Output**: 0

**Explanation**: No triplet with zero sum exists.

**Expected Time Complexity:**O(n2)  
**Expected Auxiliary Space:**O(1)  
  
**Constrains:**  
1 <= n <= 104  
-106 <= Ai <= 106

**Company Tags**

[**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Google**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Google) [**Facebook**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Facebook)

**Topic Tags**

[**two-pointer-algorithm**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=two-pointer-algorithm) [**Arrays**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Arrays) [**Searching**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Searching) [**Hash**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Hash) [**Sorting**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Sorting) [**Data Structures**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Data%20Structures) [**Algorithms**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Algorithms)

import java.util.\*;

class Triplets{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

while(t-->0){

int n=sc.nextInt();

int[] a=new int[n];

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

a[i]=sc.nextInt();

}

Solution g=new Solution();

if(g.findTriplets(a,n))

System.out.println("1");

else

System.out.println("0");

}

}

}

class Solution

{

public boolean findTriplets(int arr[] , int n)

{

int j=0;

int k=n-1;

Arrays.sort(arr);

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

j=i+1;

k=n-1;

while(j<k){

if((arr[i]+arr[j]+arr[k])==0)

return true;

if((arr[i]+arr[j]+arr[k])>0)

k--;

else

j++;

}

}

return false;

}

}