**Lemonade Change**

**Medium**

You are an owner of lemonade island, each lemonade costs **$5**. Customers are standing in a queue to buy from you and order one at a time (in the order specified by given array **bills[]**). Each customer will only buy one lemonade and pay with either a **$5**, **$10**, or**$20**bill. You must provide the correct change to each customer so that the net transaction is that the customer pays **$5**.

**NOTE:**At first, you do not have any bill to provide changes with. You can provide changes from the bills that you get from the previous customers.

Given an integer array **bills** of size **N** where **bills [ i ]** is the bill the **ith**customer pays, return**true** if you can provide every customer with the correct change, or **false** otherwise.

**Example 1:**

**Input:**  
N = 5  
bills [ ] = {5, 5, 5, 10, 20}  
**Output:** True  
**Explanation:**   
From the first 3 customers, we collect three $5 bills in order.  
From the fourth customer, we collect a $10 bill and give back a $5.  
From the fifth customer, we give a $10 bill and a $5 bill.  
Since all customers got correct change we return true.

**Example 2:**

**Input:**  
N = 5  
bills [ ] = {5, 5, 10, 10, 20}  
**Output:** False  
**Explanation:**   
From the first two customers in order, we collect two $5 bills.  
For the next two customers in order, we collect a $10 bill and give back a $5 bill.  
For the last customer, we can not give the change of $15 back because we only have two $10 bills.  
Since not every customer received the correct change, the answer is false.

//{ Driver Code Starts

// Initial Template for Java

import java.util.\*;

import java.lang.\*;

import java.math.\*;

import java.io.\*;

class CodingMaxima {

public static void main(String[] args) throws IOException {

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 obj = new Solution();

boolean ans = obj.lemonadeChange(n, a);

System.out.println(ans ? "True" : "False");

}

}

}

// } Driver Code Ends

// User function Template for Java

class Solution {

static boolean lemonadeChange(int N, int bills[]) {

int five=0;

int ten=0;

int tewnty=0;

for(int coin :bills){

if(coin==5){

five++;

}

else if(coin==10){

if(five==0){

return false;

}

five--;

ten++;

}

else{

if(ten>=1 && five==0){

return false;

}

else if(five>=3 && ten==0){

five-=3;

}

else if(five==0 && ten==0){

return false;

}

else if(ten==0 && five<3){

return false;

}

else{

ten--;

five--;

}

}

}

return true;

}

}