**Adding One**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)

Given a non-negative number represented as an array of digits, add 1 to the number ( increment the number represented by the digits ). The digits are stored such that the most significant digit is at the head of the list.

Example:

If the array has [4, 5, 6]

the resultant array should be [4, 5, 7]

as 456 + 1 = 457.

**Input:**

The first line of input contains an integer T denoting the number of test cases. The description of T test cases follows.  
The first line of each test case contains a single integer N denoting the size of array.  
The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.

**Output:**

Print the space separated resultant array in a separate line for each case.

**Constraints:**

1 ≤ T ≤ 30  
1 ≤ N ≤ 1000  
0 ≤ A[i] ≤ 9

**Example:**

Input:  
2  
4  
5 6 7 8  
3  
9 9 9  
Output:  
5 6 7 9  
1 0 0 0

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=107>

#include <iostream>

#include <stdio.h>

#include <map>

#include <string>

#include <algorithm>

#include <vector>

#define ll long long int

#include <conio.h>

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int N;

scanf("%d", &N);

std::vector<int> digits;

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

int elem;

scanf("%d", &elem);

digits.push\_back(elem);

}

//std::vector<int> res = plusOne(arr);

int u = digits.size() - 1;

digits[u]++;

while(u > 0 && digits[u] > 9){

digits[u] = 0;

u--;

digits[u]++;

}

if(digits[0] > 9){

vector<int> copia(digits.size()+1);

copia[0] = 1;

copia[1] = 0;

for(int i=1; i<digits.size(); i++){

copia[i] = digits[i];

}

//return copia;

digits = copia;

}

//return digits;

for(int i = 0; i < digits.size(); i++) {

printf("%d ", digits[i]);

}

printf("\n");

}

getch();

return 0;

}