**Sum of two numbers represented as arrays**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[Amazon](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1)

Given two numbers represented by two arrays, write a function that returns sum array. The sum array is an array representation of addition of two input arrays. It is not allowed to modify the arrays.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case contains two integers M and N separated by a space. M is the size of arr1 and N is the size of arr2.  
The second line of each test case contains M integers which is the input for arr1.  
The third line of each test case contains N integers which is the input for arr2.  
  
**Output:**

Print the sum list.  
  
**Constraints:**

1 ≤ T ≤ 100  
1 ≤ N ≤ M ≤ 1000  
0 ≤ arr1[i],arr2[i]≤ 9  
  
**Example:**

**Input:**  
2  
3 3  
5 6 3  
8 4 2  
16 4  
2 2 7 5 3 3 7 3 3 6 8 3 0 5 0 6   
4 3 3 8

**Output:**  
1 4 0 5  
2 2 7 5 3 3 7 3 3 6 8 3 4 8 4 4

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=506>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <map>

#include <string.h>

#include <math.h>

#define ll long long int

using namespace std;

int main() {

  int t;

  scanf("%d", &t);

  while(t--) {

    int M,N;

    scanf("%d %d", &M , &N);

    int num1[M];

    for(int i =0; i < M; i++) {

        scanf("%d",&num1[i]);

    }

    int num2[N];

    for(int i =0; i<N; i++) {

        scanf("%d", &num2[i]);

    }

    int i = M - 1, j = N - 1;

    std::vector<int> res;

    int me\_llevo = 0;

    while (i >= 0 && j >= 0) {

        int sum = (num1[i] + num2[j] + me\_llevo);

        res.push\_back(sum % 10);

        if (sum > 9){

            me\_llevo = (sum / 10);

        }

        else {

            me\_llevo = 0;

        }

        i--;

        j--;

    }

    while (i >= 0) {

        int sum = (num1[i]  + me\_llevo);

        res.push\_back(sum % 10);

        if (sum > 9) {

            me\_llevo = (sum / 10);

        }

        else{

            me\_llevo = 0;

        }

        i--;

    }

    while (j >= 0){

        int sum = (num2[j] + me\_llevo);

        res.push\_back(sum % 10);

        if (sum > 9){

            me\_llevo = (sum / 10);

        }

        else{

            me\_llevo = 0;

        }

        j--;

    }

    if (me\_llevo > 0){

        res.push\_back(me\_llevo);

    }

    std::reverse(res.begin(), res.end());

    for ( i = 0; i < res.size(); i++)

    {

        //Console.Write(res[i] + " ");

        printf("%d ", res[i]);

    }

    printf("**\n**");

  }

}