**Minimum sum**

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

Given an array of digits (values are from 0 to 9), find the **minimum**possible sum of two numbers formed from digits of the array. All digits of given array must be used to form the two numbers.

**Input:**

The first line of input contains a single integer T denoting the number of test cases. ThenT test cases follow. Each test case consist of two lines. The first line of each test case consists of an integer N, where N is the size of array.  
The second line of each test case contains N space separated integers denoting array elements.

**Output:**

Corresponding to each test case, in a new line, print the minimum possible sum of two numbers formed from digits of the array.

**Constraints:**

1 ≤ T ≤ 100  
1 ≤ N ≤ 30  
1 ≤ A[i] ≤ 9

**Example:**

**Input**  
1  
5  
5 3 0 7 4

**Output**  
82

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=984>

#include <iostream>

#include <stdio.h>

#include <algorithm>

#define ll long long int

using namespace std;

int main() {

   int t;

   scanf("%d", &t);

   while(t--) {

      int n;

      scanf("%d", &n);

      int arr[n];

      for(int i =0; i<n; i++) {

         scanf("%d", &arr[i]);

      }

      sort(arr, arr + n);

      ll a=0,b=0;

      //transformo a enteros los digitos alternados

      //ej: con arr = {1,2,3,4,5}

      //queda {1,3,5} y {2,4}, o sea 135 y 24

      for(int i =0; i<n; i++) {

          if(i % 2 == 0){

            a = a\*10 + arr[i];

          } else {

            b = b\*10 + arr[i];

          }

      }

      cout << a+b << endl;

   }

 // system("pause");

  return 0;

}