**Sum of diagonals**

Submissions: [1446](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2487)  Accuracy:

42.51%

   Difficulty: [School](https://practice.geeksforgeeks.org/School/0/0/)   Marks: 0

\*School Problem's Submission isn't counted in score!

Show Topic Tags   

Given a 2-D square matrix, find sum of elements in Principal and Secondary diagonals. For example, consider the following 4 X 4 input matrix.

A00 A01 A02 A03

A10 A11 A12 A13

A20 A21 A22 A23

A30 A31 A32 A33

The primary diagonal is formed by the elements A00, A11, A22, A33. And the secondary diagonal is formed by the elements A03, A12, A21, A30.

**Input:**  
The first line consists of an integer **T**i.e number of test cases. The first line of each test case consists of an integer **N**.The next line consists of **N\*N** spaced integers.

**Output:**  
Print the sum of primary diagonal elements and the secondary diagonal elements with a space in between.

**Constraints:**  
1<=T<=100  
1<=N,a[i][j]<=1000

**Example:  
Input:**  
2  
4  
1 2 3 4 4 3 2 1 7 8 9 6 6 5 4 3  
3  
1 1 1 1 1 1 1 1 1

**Output:**  
16 20  
3 3

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/sum-of-diagonals/0/?ref=self#ExpectOP) option \*\*

Contributor: Saksham Ashtputre  
[Author: saksham0751](https://auth.geeksforgeeks.org/user/saksham0751/practice/)

<https://practice.geeksforgeeks.org/problems/sum-of-diagonals/0/?ref=self>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine());

string[] input = Console.ReadLine().Trim().Split(' ');

int index = 0;

int[][] matrix = new int[n][];

int j = 0;

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

{

matrix[i] = new int[n];

for (j = 0; j < n; j++)

{

matrix[i][j] = int.Parse(input[index++]);

}

}

int d1 = 0, d2 = 0;

j = n - 1;

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

{

d1 += matrix[i][i];

d2 += matrix[i][j];

j--;

}

int sum = d1 + d2;

if (n % 2 != 0)

{

sum -= matrix[n / 2][n / 2];

}

// System.out.println(Math.abs(d1 - d2));

Console.WriteLine(sum);

}

Console.ReadLine();

}

}

}