code jam

System.out.println("hello, world!");

Round 1A 2008

A. Minimum Scalar Product

B. Milkshakes

C. Numbers

Contest Analysis

Questions asked 3

Submissions

Minimum Scalar Product

5pt	Not attempted
	2352/2567 users correct (92%)
10pt	Not attempted
	1048/2336 users correct (45%)

Milkshakes

10pt	Not attempted
	655/1042 users correct (63%)
25pt	Not attempted
	312/432 users correct (72%)

Numbers

15pt	Not attempted
	577/1925 users correct (30%)
35pt	Not attempted
	96/364 users correct (26%)

Top Scores

Bohua	100
yuhch123	100
neal.wu	100
newman	100
Plagapong	100
Ahyangyi	100
Reid	100
Qingchun	100
ploh	100
kubus	100

Problem A. Minimum Scalar Product

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input		
5 points	Solve A-small	

Large input		
10 points	Solve A-large	
	Judge's response for last submission: Correct.	

Problem

You are given two vectors $v_1=(x_1,x_2,...,x_n)$ and $v_2=(y_1,y_2,...,y_n)$. The scalar product of these vectors is a single number, calculated as $x_1y_1+x_2y_2+...+x_ny_n$.

Suppose you are allowed to permute the coordinates of each vector as you wish. Choose two permutations such that the scalar product of your two new vectors is the smallest possible, and output that minimum scalar product.

Input

The first line of the input file contains integer number **T** - the number of test cases. For each test case, the first line contains integer number **n**. The next two lines contain **n** integers each, giving the coordinates of v_1 and v_2 respectively.

Output

For each test case, output a line

```
Case #X: Y
```

where **X** is the test case number, starting from 1, and **Y** is the minimum scalar product of all permutations of the two given vectors.

Limits

Small dataset

T = 1000
$1 \le n \le 8$
$-1000 \le x_i, y_i \le 1000$

Large dataset

T = 10
$100 \le n \le 800$
$-100000 \le x_i, y_i \le 100000$

Sample

```
Input           Output

2               Case #1: -25
3               Case #2: 6
1 3 -5
-2 4 1
5
1 2 3 4 5
```

```
1 0 1 0 1
```