Problem I: Town Square

Felix J. Humble, a nerdy but rich resident of a small Midwest town, has erected 4 statues in his honor in a public park (which he happens to own). Felix is very proud of these statues, but is now worried about vandals, small children with bubble gum, and the occasional dog with a urinary tract infection. To solve this problem, he has decided to build a fence around the statues (Felix owns the local fencing company as well). For various aesthetic reasons he would like the following conditions met:

1. The enclosed area must be a square.

2. The distance between each statue and its nearest fence side should be 5 feet.

3. No two statues should have the same nearest fence side.

After working for a grand total of 12 seconds, Felix has realized that he has no idea of the length of fence needed, or even if the above conditions can be met. Since he is planning similar tributes in other parks he owns, he would like someone to write a program to solve this problem (which Felix will later take credit for, of course).

Input

The first line of the input file will contain an integer n indicating the number of test cases. The test cases will follow, one per line, each consisting of eight integer values giving the x and y coordinates for the first, second, third and then fourth statue. All values will be in units of feet, and will lie between -100 and 100. No two statues will be at the same location.

Output

For each test case, output the case number followed by one of two answers: 1) If there is a solution, output the side length of the enclosing square. If there are multiple solutions, output the side length of the solution of maximum size; 2) If it is not possible to build a square fence which meets Felix's conditions, output `no solution`. All numerical output should be rounded to the nearest hundredth of a foot. Follow the format in the examples below.

Sample Input

```
2
0 1 1 0 3 4 4 2
0 1 0 2 0 3 0 4
```

Sample Output

```
Case 1: 14.00
Case 2: no solution
```