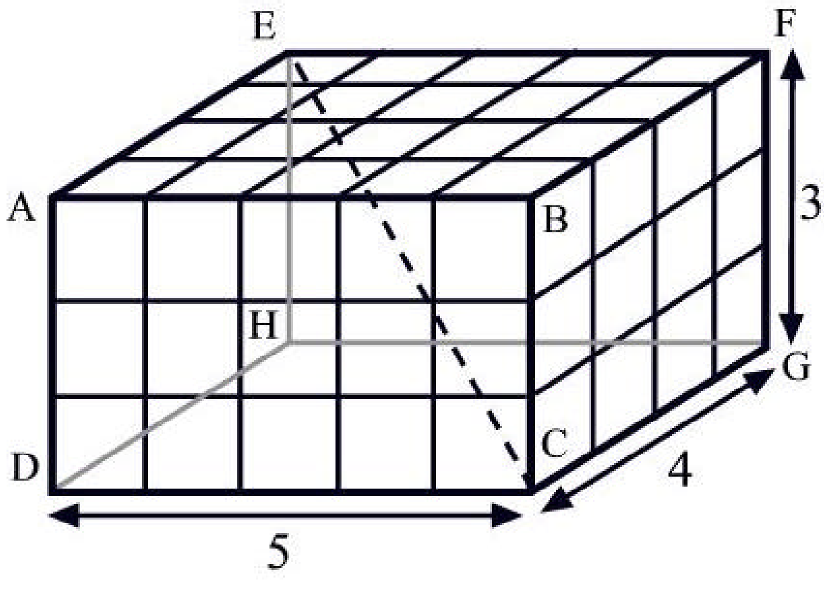
Problem D

Hyper-drive

Time Limit: 2 seconds

Hyper-drive is a term that is very frequently used in science fiction stories. Although, many believe that hyper-drive is not at all possible, many explanations and theories exist on the existence of wormholes and hyper-drives. Many say that hyper-drive is a journey through higher dimensions. In this problem we will try to calculate our cost of traveling through higher dimension based on a theory described by our old mad friend Arif. I am sure that you remember Arif. You can ask your team mates if you don't remember.
Let **P** and **Q** be two points in n-dimensional space. Let the coordinates of **P** be **(p1, p2, …, pn)** and the coordinates of **Q** be **(q1, q2, …, qn)**. The universal n-dimensional space is divided into many unit n-dimensional hyper-cubes. For visual example look at the picture below to realize how a **(5 x 4 x 3)** three dimensional universal space can be divided into **60** three dimensional unit hyper-cubes **(1 x 1 x 1)**.



Please don't ask for a visual example in higher dimension. The cost of traveling from one n-dimensional **P** point to another n-dimensional point **Q** is equal to "the number of different n-dimensional unit hyper-cubes that the straight line joining these two points passes through.". Your job is to determine this cost for two given points. For example in the previous picture the cost of hyper-drive from **C** to **E** is **10** units as **EC** passes through **10** different three dimensional hyper-cubes.

Input

The first line of the input file contains an integer **N (N<=501)** that denotes how many sets of inputs are there. The description of each set is given below: Each set starts with an integer **D (0&ltD<=10)** which denotes in what dimension we want to measure the cost. Each of the next two lines contains **D** integers. The **D** integers in the first line denote the coordinates for **P** and the **D** integers in the second line denotes the coordinates for **Q**. All these integers will be positive and fit within **32-bit signed integer**.

Output

For each set of input produce one line of output. This line contains the serial of the output and then the cost of traveling from **P** to **Q**, which will obviously be an integer. Look at the sample output below for details.

Sample Input	Sample Output

`2`
`2`
`10 10`
`10 13`
`1`
`10`
`20`
	`Case 1: 0`
	`Case 2: 10`

Problemsetter: Shahriar Manzoor, special thanks to Derek Kisman

Note: In the first sample output for sample input the cost of traveling from (10, 10) to (10, 13) is shown zero because the line connecting these two points does not really enter any two dimensional hyper-cube. The line just goes along the edges of the two dimensional hyper-cubes.

Problem E

Traveling Politician

Time Limit: 2 seconds

A politician from the Alliance of Conservative Monarchists (**ACM**) is campaigning for the next election. In order to guarantee his victory, he has to make at least k public speeches. He will give one speech every day. If he has to give several speeches in the same city, they cannot be on consecutive days because that would be unproductive. However, the politician believes that giving speech in one day's interval is not useless; for example, giving one speech on Monday and the next one in the same city on Wednesday is alright because after two days, the people will forget about his first speech and his second speech will have as much effect as the first one.

He is absolutely certain that he will win, so at the same time he is moving to the capital. This means that his first speech will be given in his hometown, and his last speech - in the capital city. He knows that his speech-giving abilities deteriorate when he is tired. So he does not want to give more speeches than he has to; k speeches will be enough to win. What is the minimum number of speeches he has to give in order to start in his hometown, end up in the capital and give at least k speeches (Including his speeches in his hometown and in the capital) on the way, without ever giving a speech in the same city on two consecutive days?

Input

The input will consist of several test cases. Each test case will begin with **3** integers on a line - **n** (the number of cities on the map), **m** (the number of roads connecting cities) and **k** (the minimum number of speeches). The next m lines will each contain **2** integers, **u** and **v**, meaning that the politician can visit city v immediately after visiting city u. All other routes of travel are infeasible from the point of view of his budget. The politician's hometown is city number **0**, and the capital is city number **n-1**. You can assume that **2<=n<=50** and **2<=k<=16**. Input is terminated by a line containing three zeroes.

Output

Print one line per test case, giving the minimum total number of speeches. If this is impossible to do, print "**LOSER**". See examples. If in a scenario the politician requires to give more than **20** speeches he should be considered a LOSER and so in that case you should print "**LOSER**" as well.

Sample Input	Sample Output
3 3 3 0 1 0 2 1 2 5 6 5 0 1 0 3 1 2 2 4 3 2 3 4 3 3 10 0 1 1 0 1 2 0 0 0	3 LOSER 11

Problemsetter: Igor Naverniouk, special thanks to Shahriar Manzoor

Problem F
Smallest Bounding Rectangle

Input: standard input
Output: standard output
Time Limit: 3 seconds

Given the Cartesian coordinates of n (> 0) 2-dimensional points, write a program that computes the area of their smallest bounding rectangle (smallest rectangle containing all the given points).

Input

The input file may contain multiple test cases. Each test case begins with a line containing a positive integer n (< 1001) indicating the number of points in this test case. Then follows n lines each containing two real numbers giving respectively the x- and y-coordinates of a point. The input terminates with a test case containing a value 0 for n which must not be processed.

Output

For each test case in the input print a line containing the area of the smallest bounding rectangle rounded to the 4th digit after the decimal point.

Sample Input

```
3
-3.000 5.000
7.000 9.000
17.000 5.000
4
10.000 10.000
10.000 20.000
20.000 20.000
20.000 10.000
0
```

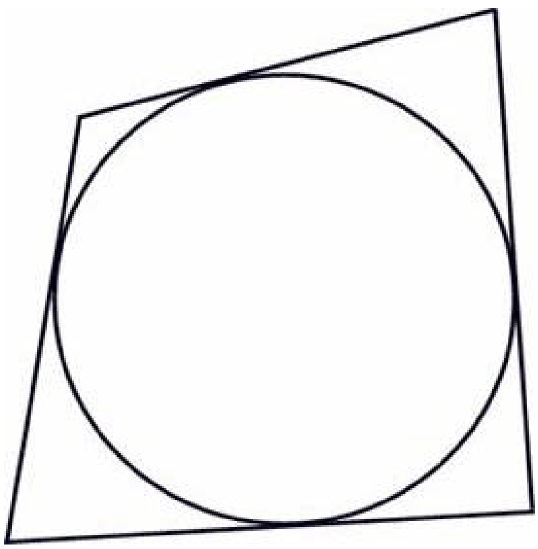
Sample Output

```
80.0000
100.0000
```

Rezaul Alam Chowdhury

Problem G
Maximal Quadrilateral
Time Limit: 1 second

A circle is inscribed in a quadrilateral and the circle touches all four sides of it. Given the perimeter and the length of two adjacent sides of the quadrilateral, you will have to find the maximum possible radius of the circle. You can assume that **pi=2*cos-1(0.0)**.



Input
The first line of the input file contains an integer **N (N<=100)** that indicates how many sets of input are there. Each of the next **N** lines contains three integers **P**, **A**, **B**. Here **P** is the perimeter and **A**, **B** are the length of two adjacent sides of the quadrilateral.

Output
For each line of input except the first one, produce one line of output. This line should contain the serial of the output and then the maximum possible radius of the circle. The radius should have six digits after the decimal point. If the formation of such a quadrilateral is impossible then print the line "**Eta Shombhob Na.**" (Sorry! It is a sentence in Bangla, which means "*This is impossible.*"). Errors less than **0.00001** will be ignored. The lengths of all sides of a valid quadrilateral should be positive. Look at the output for sample input for details:

Sample Input	Sample Output

`2`
`20 5 6`
`20 10 12`
	`Case 1: 2.449490`
	`Case 2: Eta Shombhob Na.`

Problemsetter: Shahriar Manzoor, special thanks to Derek Kisman