

Problem C. Communication Between Robots

Input file: *standard input*
Output file: *standard output*
Time limit: 7 seconds
Memory limit: 512 mebibytes

In the year 21xx, human beings are proliferating across the galaxy. Since the end of the last century, thousands of pioneer spaceships have been launched in order to discover new habitable planets.

The Tripenser is one of those spaceships, heading toward the Andromeda galaxy. After a long, long cruise in the hyperspace, the crew has finally found a very hopeful candidate planet. The next thing to do is to investigate the planet to find out whether it is really suitable for new residents or not.

For that purpose, the ship is taking some unattended landers. The captain Juclean Dripac decided to drop them to the planet and collect data about it. But unfortunately, these robots are a bit old and not very clever, so the operator has to program what to do on the planet before the landing. Many crew members including you are called for making this important plan.

The most complicated phase in the mission is to gather and integrate all the data collected independently by many robots. The robots need to establish all-to-all communication channels to exchange the data, once during the mission. That is, all the robots activate their communication channels all together at the predetermined moment of time, and they exchange the data with each other once, at that exact moment. This one moment can happen at the very beginning of very end of the mission, or any time in between. You have to choose this moment, and the operator will program the robots accordingly.

The robots use wireless channels to communicate with each other, so the distance between robots does not limit the connectivity. But the farther two robots are, the more power they have to use for communication. Due to the limitation of the battery capacity, you want to save the transmission power as much as possible.

On the bright side, communication units of the robots also have routing functionality, so robots can talk indirectly using other robots as relays. Consider a graph where vertices represent robots and edges represent communication channels established between them. If the graph is connected, all-to-all communication can be established.

Your task is to write the program to calculate the minimum total transmission power required for all-to-all communication among the robots. Remember that they only have get one communication session at some moment during the mission, and you can choose this moment in advance. Each robot moves linearly on the planet surface. The transmission power required for two robots is proportional to the distance between them, so the cost here is exactly the sum of the distances between each pair of robots which establish a communication channel.

You may also regard the planet surface as a two-dimensional plane, as it is huge enough. The time required for communicating data among robots is also negligible.

Input

The input contains no more than 200 datasets. Each dataset has the format shown below.

$N T$

$x_1 y_1 vx_1 vy_1$

...

$x_N y_N vx_N vy_N$

The first line of each dataset contains two integers; N is the number of robots used for gathering data ($2 \leq N \leq 16$), and T is the time limit of the mission ($1 \leq T < 1000$).

Each of the following N lines describes the motion of a robot. (x_i, y_i) and (vx_i, vy_i) are the initial landing position and the velocity of the i -th robot, respectively ($|x_i|, |y_i| < 10^6$, $|vx_i|, |vy_i| < 1000$).

The last dataset is followed by a line containing two zeros. This line is not a part of any dataset and

should not be processed.

It is guaranteed that the sum of all N in the datasets does not exceed 1700.

Output

For each dataset, output in a line the minimum communication cost required for all-to-all communication. Your program may output an arbitrary number of digits after the decimal point. The absolute error should be less than or equal to 0.001.

Example

standard input	standard output
4 2	6.00000000
2 0 0 1	4.24264069
0 4 1 0	
4 6 0 -1	
6 2 -1 0	
4 6	
2 0 0 1	
0 4 1 0	
4 6 0 -1	
6 2 -1 0	
0 0	