Problem B: Random Walk

Input: `randomwalk.in`
Output: `randomwalk.out`

Random walks are used to model a wide range of phenomena, from Brownian motion to gambling. For example, a gambler who bets on heads or tails on a coin toss wins or loses his bet each turn. The amount of money the gambler has throughout the game is a random walk. Although the bets in each turn may be different, it is easy to see that the gambler wins the maximum amount of money if he wins every turn. Similarly, he loses the maximum amount if he loses every turn.

We consider the following two−dimensional variation of the random walk. We are given n two−dimensional nonzero vectors $v_i = (x_i, y_i)$, no two of which are parallel. In step i, a coin is flipped. If it is heads, we move x_i meters in the x direction and y_i meters in the y direction. If it is tails, we move $-x_i$ and $-y_i$ meters in the x and y directions.

We are interested in computing the maximum distance we can be away from our starting point. This is easy in one−dimension, but it is not so easy in the two−dimensional version.

Input

The input consists of a number of test cases. Each test case starts with a line containing the integer n, which is at most 100. Each of the next n lines gives a pair of integers x_i and y_i specifying v_i. The coordinates are less than 1000 in magnitude. The end of input is specified by $n = 0$.

Output

For each test case, print on a line the maximum distance we can be away from the starting point, in the format shown below. Output the answer to 3 decimal places.

Sample input

```
3
1 1
0 1
−1 1
2
4 0
1 1
7
1 3
−2 −7
7 8
−2 9
−7 −3
4 −3
−2 −2
0
```

Output for sample input

```
Maximum distance = 3.000 meters.
Maximum distance = 5.099 meters.
Maximum distance = 37.336 meters.
```