Problem Statement (TurtleSpy)

The Very Heterogeneous State of Feudalia's army is designing a new spy robot. The robot is disguised as a turtle and takes four different commands:

- "right X", where X is a integer between 1 and 359, inclusive. Will add X degrees to the turtle's direction angle.
- "left X", where X is a integer between 1 and 359, inclusive. Will subtract X degrees from the turtle's direction angle.
- "forward X", where X is a integer. Will move X units in the direction given by the turtle's direction angle.
- "backward X", where X is a integer. Will move X units in the opposite direction of the turtle's direction angle (The angle minus 180 degrees).

The army created a program that made the robot infiltrate deeply into foreign land by making the robot terminate as far as possible from the original location. Unfortunately, Feudalia is world famous for the ineptitude of its public officials. The program got mixed up and there is no way to tell the original order of the commands. You are given a String[] **commands**. Each element of **commands** represents a command in the described format. Find the order for all of the commands in the input that will maximize the Euclidean distance between the point where the robot was started and the point it reached after all commands were executed. Return the maximum distance.

Definition

Class:	TurtleSpy
Method:	maxDistance
Parameters:	String[]
Returns:	double
Method signature:	double maxDistance(String[] commands)

(be sure your method is public)

Notes

-The Euclidean distance between the points (x1,y1) and (x2,y2) is sqrt((x2-x1)^2 + (y2-y1)^2).-Your return value must be within 1e-9 relative or absolute error of the correct value

Constraints

-**commands** will contain between 1 and 50 elements, inclusive.-Each element of **commands** will be a string in the form "COMM X", where COMM is one of "left", "right", "forward" or "backward" (quotes for clarity) and X is a integer between 1 and 1000, inclusive.-For each element of **commands**, if the command is "left" or "right" then X will not exceed 359.

Examples

0)

```
{"forward 100", "backward 100", "left 90" }

Returns: 141.4213562373095
```

That distance is possible if we make sure to run the *"left 90"* command inbetween the other two commands.

1)

```
{"left 45", "forward 100", "right 45", "forward 100"}

Returns: 200.0
```

We can, for example, run the commands in the following order: "forward 100", "left 45", "right 45" and "forward 100".

2)

```
{"left 10", "forward 40",  "right 30", "left 10", "backward 4", "forward 4" }

Returns: 40.58520737741619
```