

Homework #2 - Robot & Treasures

After finishing the Simple Robot Controller, you are able to move the robot on the 2D plane right-and-left. However, your boss are not satisfied with your work, he asks you to modify the moving method and to add something new to the controller. First, instead of moving the robot by giving the moving distance along x-axis and y-axis, the new controller should receive the moving direction, and just one number stands for the moving distance. Second, the new controller should have the ability to detect whether the giving operation will make the robot moving out of range or not. Once a operation moves the robot out of range, it is considered an illegal one, the controller should show the warning message, and the robot should just stay at the original position. Third, there are some treasures scatter on the plane, their positions are given before you starting moving the robot. When the robot moving on or through a treasure, the controller should show a message, and the robot picks the treasure up. Finally, in addition to show the moving path of the robot as you have done in your last work, the controller should also show how many treasures does the robot get during the traveling.

Input Format

The first three lines are the same as your last work.

Each of the first two lines contains a **signed** integer, x0 and y0 ($-10000 \leq \underline{x0}$, $\underline{y0} \leq 10000$), means the start position of the robot in x-axis and y-axis, respectively.

The third line contains an **unsigned** integer n ($1 \leq \underline{n} \leq 20$), means there are how many operations to be given.

The Fourth line contains an **unsigned** integer Tn ($1 \leq \underline{Tn} \leq 30$), means there are how many treasures on the plane.

The following $2\underline{Tn}$ lines are the positions of the treasures, two lines a pair, each containing two **signed** integers Txi and Tyi ($-10000 \leq \underline{Txi}$, $\underline{Tyi} \leq 10000$).

The last $2\underline{n}$ lines are moving operations, two lines a pair. The first line of each pair contains a string("Up" , "Down" , "Left" or "Right"), means the moving direction, while the second line contains an **unsigned** integer d ($0 \leq \underline{d} \leq 20000$), means the moving distance.

Output Format

After inputting the first $4 + 2T_n$ lines, show the start position of the robot with the format as follows:

"The robot starts at (x, y)."

Then, for each inputting two lines(means a moving operation), compute the new position of the robot after moving. If it is out of range, do not move the robot, and show the warning message as follows:

"Warning! This is an illegal operation."

If it is not out of range, move the robot, and show the treasure(s) you get(if any) with the format as follows:

"The robot gets k treasure(s) at: (x0, y0) (x1, y1) ... (xk, yk)"

No matter the robot moves or not, show the current position with the format as follows:

"The robot is now at (x, y)."

After all the operations are completed, show the moving path list of the robot with the format as follows:

"The moving path of the robot: $(x_0, y_0) \rightarrow (x_1, y_1) \rightarrow \dots \rightarrow (x_n, y_n)$."

But if there are some operations does not move the robot(it could be a zero moving distance or an illegal move), your should not consider it as a move in the moving path list.

Finally, show how many treasure(s) do you get with the format as follows:

"The robot gets n treasure(s) in total."

Note

1. The range of the plane is -10000 to 10000 in both x-axis and y-axis.
2. You can assume there are no treasures at the start position of the robot.
3. You can assume there are no two or more treasures at the same position.
4. The order of the treasures in the treasure list is not cared.
5. A treasure can be picked up only once.
6. Be sure your moving path list should **not** contain the terms such as $(x_i, y_i) \rightarrow (x_j, y_j)$, while both x_i equals to x_j and y_i equals to y_j .

Sample I/O

0 // Input, the start position of the robot in x-axis.

0 // Input, the start position of the robot in y-axis.

3 // Input, there are 3 operations below.

3 // Input, there are 3 treasures on the plane.

1 // Input, the x-axis of the 1st treasure.

0 // Input, the y-axis of the 1st treasure.

-1 // Input, the x-axis of the 2nd treasure.

0 // Input, the y-axis of the 2nd treasure.

2 // Input, the x-axis of the 3rd treasure.

0 // Input, the y-axis of the 3rd treasure.

The robot starts at (+0, +0).

// Your output, the start position of the robot.

Right // Input, the direction of the 1st operation.

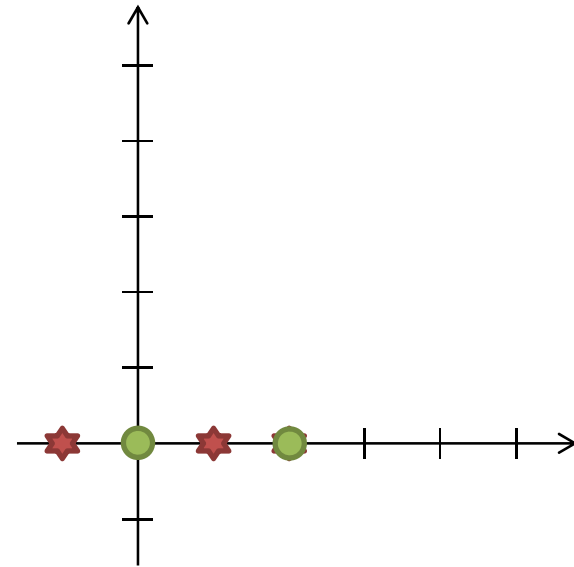
2 // Input, the distance of the 1st operation.

The robot gets +2 treasure(s) at: (+1, +0) (+2, +0)

// Your output, the treasures your get during the movement.

The robot is now at (+2, +0).

// Your output, the position of the robot after the 1st operation.



Right // Input, the direction of the 2nd operation.

20000 // Input, the distance of the 2nd operation.

Warning! This is an illegal operation.

// Your output, show the warning message if the robot will out of range.

The robot is now at (+2, +0).

// Your output, the position of the robot after the 2nd operation.

Up // Input, the direction of the 3rd operation.

3 // Input, the distance of the 3rd operation.

The robot is now at (+2, +3).

// Your output, the position of the robot after the 3rd operation.

// Notice! No treasure list if there is no treasure found.

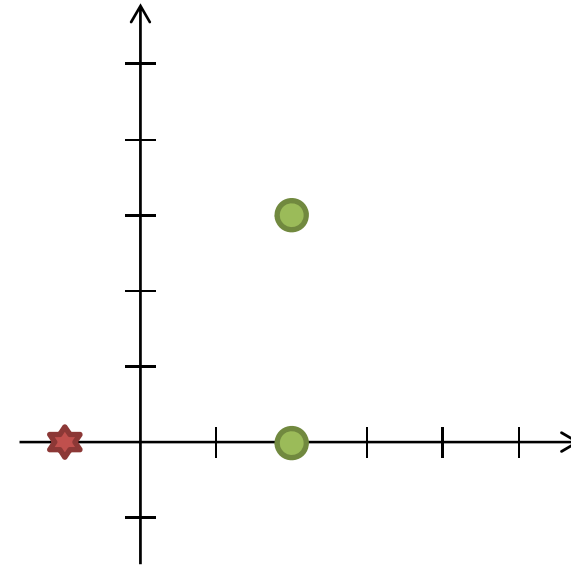
The moving path of the robot: (+0, +0) -> (+2, +0) -> (+2, +3).

// Your output, the moving path of the robot.

// Notice! The 2nd operation does not form a movement.

The robot gets +2 treasure(s) in total.

// Show how many treasures you get.



Requirements

1. The decision directives in chapter 6 are **NOT** permitted to use.
2. You need to add some comments in your source code.
3. Write a report no more than one page to share how you have done your homework and problems you experienced.
4. Upload your source code(.asm file) and report (in .doc or .pdf format) to the E3 platform.
5. The deadline is 2011/4/15(Fri.) 23:59:59, you can have late homework before 2011/4/19(Tues.) 23:59:59 with 15% discount per day, after that you will only get **ZERO**.
6. Please **DO NOT** take a copy from others, or you will only get **ZERO**.

Hint

1. Design some procedures to do something such as to detect whether the robot out of range or to find the treasures, etc.
2. If the destination is too far for the loop instruction to jump, try to use the conditional jump instructions instead.
3. Try to optimize the logic of finding the treasures.