Assembly Rover

1 Objective

The purpose of this test is to enable you to demonstrate your proficiency in solving problems using software engineering tools and processes.

Read the specification below and produce a solution. Your solution should be in the form of completed code.

If you are unsure of any part of the specification, then you should ask to speak to a member of the Software Engineering team either via your recruitment agent or HR contact.

The problem specified below requires a solution that receives input, does some processing and then displays some output. You can use either a graphical user interface or a command line interface for the input and output of your solution. You should provide sufficient evidence that your solution is complete by, as a minimum, indicating that it works correctly against various input data. Using a unit testing framework would satisfy these requirements.

The interviewer will be interested in:

1. Your ability to read and interpret the specification below.

2. The architectural design of your solution.

3. The readability of your code.

4. Your overall approach to this exercise.

2 Specification

A set of numbered components are placed on a grid, a rover is placed on a random grid coordinate and has to pick up the numbered components in ascending order (starting with 1). Once the last component has been picked up the solution is complete.

The grid is a square with origin (0,0) in the bottom left corner.

The rover can move only one grid-unit in any of the four cardinal directions (North, South, East, West) per move. There are no limits on how many times the grid can be traversed.

The rover can only detect a component if the rover and the component share the same grid coordinates.

The rover is aware of both the grid size as well as its own location within it.

The components can only be picked up in ascending order, if no components have been picked up only the first one (1) can be picked up. If components 1, 2, and 3 have been picked up, the next component eligible for pickup is 4.

See **Figure 1** for a sample of a size 2 grid.

The solution should be created as a simulation of the rover as it performs the task set out by the specifications and the output should reflect the movements performed by the rover. We only require one solution to be found, we're not asking for an exhaustive search of all possibilities.

2.1 Input

The input should consist of the following values:

1. The grid size.
2. The number of components.
3. The location of the components (numbered 1, 2, 3, 4, … , *N*) on the grid.
4. The starting position of the rover.

2.2 Output

The output should be represented as the steps taken by the rover to accomplish the task, this can be as simple as a list of moves/actions or as elaborate as an interactive animated sequence. How the data is represented has little to no impact on the judging of the exercise, the candidate should choose something they are familiar and comfortable working with to reduce the time it will take to implement.

3 Sample

This is a trivial example meant to illustrate the desired operational behavior.

3.1 Input

1. Grid size: 2
2. Number of components: 2
3. Component locations:
    a. Component 1: 1,1
    b. Component 2: 0,0
4. Rover position: 1,0

Which would result in the following grid setup:

	C1
C2	R

FIGURE 1.

3.2 Output

Note that this is simply an example of a way of representing the output, all that is required is that the output is clear and concise.

N, S, E, and W represent directional movement, P represents component pickup.

For this particular sample the two shortest valid solutions are WNEPSWP and NPWSP.

3.3 Additional Sample Grids

3.3.1 Size 3 grid, 2 components

R		
C2	C1	

FIGURE 2.

3.3.2 Size 4 grid, 3 components

C3			
		C2	
	R		
	C1		

FIGURE 3.

3.3.3 Size 8 grid, 5 components

				R		C2	
C4							
					C1		
					C5		
	C3						

FIGURE 4.