**Code Exercise: A-Maze-ingly Retro Route Puzzle**

This code exercise is (very loosely) based on some of the concepts in old school [text adventure](http://en.wikipedia.org/wiki/Text_adventure) games.

In this exercise you are supplied with two files. The first is an XML document (with inline DTD) that describes an adventure game map. It will look something like this:

**<map>**

**<room id="1" name="Hallway" north="2" />**

**<room id="2" name="Dining Room" south="1" west="3" east="4"/>**

**<room id="3" name="Kitchen" east="2">**

**<object name="Knife"/>**

**</room>**

**<room id="4" name="Sun Room" west="2">**

**<object name="Potted Plant"/>**

**</room>**

**</map>**

As you can see, a room may or may not permit travel in one of the four cardinal directions and may or may not contain "objects". The second file is a plain text file where the first line indicates the ID of the room the player starts in, and each subsequent line lists the name of an object they must collect. This file will look something like this:

**2**

**Potted Plant**

**Knife**

The objective is to write a program that will:

* Parse the XML and create a model of the map
* Read the plain text file, noting which room to start in and which items must be collected
* Output a valid route one could take to collect all the items specified in the text file

Given the above example the following is (one of the potentially) correct outputs:

**ID  Room          Object collected**

**----------------------------------**

**2 Dining Room None**

**1   Hallway       None**

**2   Dining Room   None**

**3   Kitchen       Knife**

**2   Dining Room   None**

**4   Sun Room      Potted Plant**

A more sophisticated output would mimic the journey of the imaginary player exploring the maze, e.g.

**In the Dining Room**

**I go south**

**In the Hallway**

**I go north**

**In the Dining Room**

**I go west**

**In the Kitchen**

**I collect the knife**

**I go east**

**In the Dining Room**

**I go east**

**In the Sun Room**

**I collect the Potted Plant**

Make sure you pay attention to whether your moves are actually valid in the context of what is described in the map.xml file.

You may have some questions you'd like to ask to clarify the above requirements, e.g. "Does my output have to be exactly like the example? Do I have to find the optimal route or simply a valid route?" We’re unlikely to provide any further clarification, but instead would ask you to make and state your assumptions as part of completing this exercise.