CSCI 3005 – Programming Assignment 2 – Spring 2018

The International Wildlife Management Organization (IWMO) supervises the transport of various species of animals between zoos. Frequently, various species must be shipped from one zoo to another. To minimize expenses, the IWMO tries to use as few containers as possible, but there are important restrictions on the species that can be transported together:

- Carnivores and herbivores cannot be transported together, unless they are both domesticated
- Any animal which is part of the endangered species list must be transported by itself, since the IWMO cannot take the risk of possible disease transmission if transported in close quarters with other animals
- Animals which are in the "near threatened" or "vulnerable categories" cannot be transported with any carnivores (just in case the carnivore gets really hungry …)

You are asked to write software that, given a list of animals to be transported, finds the smallest number of containers needed to complete the shipment. Evidently, this problem can be attacked using the graph-coloring backtracking algorithm studied in class. Keep in mind that, although the scheduling restrictions will allow your program to prune the search tree significantly, the solution can be very computationally intensive if the number of animals being shipped is large. Your solution is to be implemented as a class named AnimalTransport containing the following public methods:

`AnimalTransport(String filename)`: a constructor to read in the data from a text file**,** which contains information about every species in the IWMO inventory. The file is in the .csv (comma-separated values) format and contains one line per species, including the following fields:

- *Species name*
- *Classification: carnivore, herbivore, insectivore, or omnivore*
- *Order*
- *Conservation status, expressed as a two-letter code: en (endangered), vu (vulnerable), nt (near threatened), cd (conservation dependent), lc (Least Concern), do (domesticated), ne (not evaluated)*
- *Adult body weight (in kilograms)*

A sample file named iwmo1.csv has been provided for testing. However, note that your class should handle files of any size, as long as the data is in the expected format.

`int minContainers(String animals)`: returns the minimum number of containers needed to transport the animals in the string passed as argument. The string contains a series of species names separated by commas.

`String getListing(String animals)`: returns ONE sample transport listing that utilizes the minimum number of containers needed to transport the animals in the string passed as argument. For convenience, you may simply label the containers: Container 1, Container 2, …, Container n, followed by the names of the species to be included in the container. The precise String format is left to you, but be sure everything is labeled and formatted attractively.

The IWMOTest.java program and sample text files are available for partial testing. Submit your AnimalTransport.java (and any other .java files developed as part of your solution) to Mimir for testing.