Representation & IO Team

homework for applicants

Create a Java library with the following specification:

- **Input (short notation)**:
  - A text file which describes an undirected graph.
  - Allowed characters: `[A-Z\-\.]`
  - Whitespaces are not allowed, the input is inline.
  - Syntax:
    - The vertices are marked as capital letters.
    - Two adjacent letters are connected with an undirected edge
    - An undirected edge can be implicit, see the rule above, or explicit by putting a `-` character between the two letters. `AB` and `A-B` means the same.
    - A `.` means that two adjacent nodes are not connected
  - Semantics:
    - The same letters means the same vertices
- Required API functionalities:
  - Read a graph from the short notation into a data structure
  - Get the spanning forest of the graph
  - Get a set of fundamental cycles in the graph
  - Convert a graph to a string. This string should be the human readable description that contains the vertices, edges, and number of fragments for the graph.
- Also add a Main class with a main method that converts a graph given in a short notation to human readable form and also prints out the following details:
  - Number of fundamental cycles in the graph
  - The number of edges in the spanning forest of the graph
- **You are allowed to use 3rd party libraries**, but it is not mandatory.
- Nice to have:
  - Unit and acceptance tests
  - Apidoc comments
  - +1 API functionality: Get the short notation for the implemented data structure (if possible)

Applying the solution:

You can shoot your questions and send your solution in a zip archive or as a link to an online source in an e-mail.
E-mail address: iori@chemaxon.com, galexics@chemaxon.com, zkuti@chemaxon.com, isolt@chemaxon.com, atomin@chemaxon.com, ekardos@chemaxon.com