Integrative practice
Part 1
Computational mathematics

Professor: Gilberto Huesca Juárez

Name: _____

Student ID: _____

Read the assignment before start
In Canvas, write the names and student IDs of all the team members, and the group
This activity is in pairs
Make the proposed activities and at the end, if there are programs to deliver, upload the files packaged within a ZIP file.
You may use java, python or C++.
The name of the main file must be Main.xxx where xxx is the extension for your program file.
The name of the ZIP file must be Exxxxproject1.zip where xxxxx is your student ID. For example, if your student ID is 123456, the file name must be E123456project1.zip
The programs must be properly commented (for example, if you do it in Java, your program should use JavaDoc).
LATE DELIVERY NOT APPLICABLE.
This activity counts as 20% of the final grade.

Make a program that reads from a file the elements that define an DFA-λ and then, shows the processing of a string, step by step.

The transition table will be defined in a txt file. The file shall be defined as follows:

- The first line indicates the set of states of the automata separated by commas.
- The second line indicates the alphabet symbols separated by commas
- The third line indicates the initial state
- The fourth line indicates the set of final states separated by commas.
- The following lines indicate the transition table in the following format:

state, symbol = > states

states is a list of elements separated by commas.

Example, the following line

q0, a => q1,q2

indicates that the DFA processes the following: $\delta(q0,a) = \{q1,q2\}$

To indicate a spontaneous transition, the word “lambda” will be used.

It is not necessary that all transitions are specified in this file. An evaluation may not appear if a state indicating that the result of that evaluation is the empty set

The program should print step by step the processing. And this should be repeated while the user wants to.

The test files are only examples. Your program should work for any file that follows the conventions for the **structure** denoted above. Use functions as "split" to process the lines in the file instead of read char by char. This will make your program more general. You program will be tested with files different to those used as examples.

Your program should have at least three methods:

1. Lambda closure. It should receive a set of states. It should return a set of states.
2. Transition function. It should receive a state and a character. It should return a set of states.
3. Extended transition function. It should receive a state and a string. It should return a set of states.

The program should use recursion correctly at least for the extended transition function.

Your program should show a correct use of data structures. At least, the transition table should be represented as Dictionary {key: state, value: Dictionary{key: char or lambda, value: list_of_states}}

Your program should apply correctly the conceptual elements of this course.

Do not worry about validating the values in the input file. Suppose that were built correctly.