CSE 211: Discrete Mathematics **(Due: 26/11/21)**

Homework #2

Instructor: Dr. Zafeirakis Zafeirakopoulos
Assistant: Gizem Süngü

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- It is not a group homework. Do not share your answers to anyone in any circumstance. Any cheating means at least -100 for both sides.
- Do not take any information from Internet.
- No late homework will be accepted.
- For any questions about the homework, send an email to gizemsungu@gtu.edu.tr
- The homeworks (both latex, pdf and/or source code files in a zip file) will be submitted into the course page of Teams.
- The latex, pdf or source code and zip files of the homeworks should be saved as "StudentId".{tex, pdf, c, py, cpp, java, zip}.
- If the answers of the homeworks have only calculations without any formula or any explanation -when needed- will get zero.

Problem 1: Functions (60 points)

Write an algorithm to determine if a given function $f : A \to B$ for the given sets A and B, is onto, one-to-one or bijective.

Your code should meet the following requirements, standards and tasks:

- Read the sets A and B in the text files "input_%d.txt" where d = $\{1, 2, 3\}$.
- **Example Input:**
  - A
  - apple
  - yellow
  - 4
  - B
  - orange
  - orange
  - 2
- The first line says "A".
- The following lines give each element of A.
- Each element in the set A can be a number or a string whereas they cannot be letters (since the letters represent sets).
- Read each line as an element of set A until reading the line where "B" is written.
- When you read the line "B", the following lines give each element of set B and they have the same rules with the elements in set A.

- In order to build the function $f : A \to B$, match each two elements from A to B based on their reading orders. For example, the first element in set A is the pre-image of the first element in set B.
  - apple → orange
  - yellow → orange
  - 4 → 2
- Your algorithm is responsible to detect if the function is one of them:
  - onto but not one-to-one
  - one-to-one but not onto
  - bijective
  - neither onto nor one-to-one
  - not a function (if it is not a function you don't need to detect as neither onto nor one-to-one.)
- Print the result on terminal. In the example, you should print "Problem 1: onto but not one-to-one."
- Any programming languages that you prefer are allowed.
- The input file should be given as an argument on the command. Your code should require the following commands to compile and run it:
  - **Java:**
    - Compile: javac hw2.java
    - Run: java hw2 input_file_name.txt
  - **Python:**
    - Run: python (or python3) hw2.py input_file_name.txt
  - **C:**
    - Compile: gcc hw2 -o hw2.c
    - Run: ./hw2 input_file_name.txt
  - **C++:**
    - Compile: g++ hw2.cpp -o hw2
    - Run: ./hw2 input_file_name.txt
- The main file of your code should be named as your student id.
- Write comments on your each function to describe its aim.

Problem 2: Functions (10 points)

Your code is available to detect if f^{-1} is one of them:

- onto but not one-to-one
- one-to-one but not onto
- bijective
- neither onto nor one-to-one
- not a function (if it is not a function you don't need to detect as neither onto nor one-to-one.)

Example: "Problem 2: not a function"

Problem 3: Functions (30 points)

Extend your algorithm in Problem 1 to detect if a function composition is onto, one-to-one or bijective. For instance, extend your input as follows:

- A
- apple
- yellow
- 4
- B
- orange
- orange
- 2
- C
- 3
- abc
- black
- D
- def
- 234
- 56

In the example let $f_1 : A \to B, f_2 : B \to C, f_3 : C \to D$ be functions, then your algorithm should say if $f_3 \circ f_2 \circ f_1$ is one of them:

- onto but not one-to-one
- one-to-one but not onto
- bijective
- neither onto nor one-to-one
- not a function (if it is not a function you don't need to detect as neither onto nor one-to-one.)

Consider that the number of functions in the composition can be vary so the number of sets in the input file can be vary. Example: "Problem 3: not a function"