COMP 482 Project 1: Checking Stability

Due: Tuesday February 15 2355

Points: 30 points possible

Overview: Your program will be given an instance of StableMarriage and a matching and you will determine how many instabilities it has. Note a stable matching will output 0 and an unstable matching will output an integer greater than 0.

Details: The input will come from a file called input.txt which will be placed in the same directory as your java file. The first line of the file will have a single integer value N which will be the number of men (or women since the number of men equals the number of women). The next N lines will be the whitespace separated preference lists of the N men (ie each of the next lines will be a permutation of 1, 2, ..., N). The next N lines will be the whitespace separated preference lists of the N women. The next N lines will be a whitespace separated matching. See the sample input below for examples.

Your program will determine the number of instabilities and output this value.

The obvious way to way to design the program is to check every man/woman pair to determine whether it is an instability (when both the man prefers the woman to his partner and the woman prefers the man to her partner). This is acceptable. There are simple ways to speed up the process. You may use any technique you understand.

You can discuss the algorithm to be used with anyone and consult any source (books, internet, etc). However, for this project, you are expected to write the code on your own with limited or no assistance from the professor, no assistance from others, and limited or no assistance from other sources (books, internet, etc). To clarify, you can seek assistance in understanding the task, but "your code" should be written by you: not written by others, not copied from others, not copied from books/internet.

Picky, but required specifications: Your project must:

- be submitted via canvas.
- consist of 1 or more dot-java files (no class files, zip files, input files or other files should be submitted). Each file must have your name and which project you are submitting as comments on the first 2 lines.
- not be placed into any package (for the java pedants, it must be in the default package).
- be designed and formatted reasonably (correct indentation, no excessively long lines, no excessively long methods, has useful method/variable names, etc)
- have one file called Project1.java.
- compile using the command 'javac Project1.java'.
- run using the command 'java Project1',
- accept input from a file called input.txt in the same directory as the java file(s) formatted precisely as described above.
- accomplishes the goal of the project. In other words, the output should be the correct answer, computed in a valid way, formatted correctly.
- be submitted on time (early and multiple times is fine).

For each listed item that you fail to follow, expect to lose at least 5 points. In particular, submitting via anything other than canvas will result in a 0 and submitting more that a week late will also result in a 0.

Sample execution: If input.txt contains

```
3
1 2 3
2 3 1
3 1 2
3 2 1
1 3 2
2 1 3
1 2
2 3
3 1
```

then the output should be just the number of instabilities

```
0
```

More detailed explanation of sample input/output:
The input file describes an instance with the preference lists

M1	W1	W2	W3
M2	W2	W3	W1
M3	W3	W1	W2

W1	M3	M2	M1
W2	M1	M3	M2
W3	M2	M1	M3

and the matching $\{(M1, W2), (M2, W3), (M3, W1)\}$. This matching is stable, since every woman is paired with her favorite man (a sufficient, but not necessary condition). However, your code would likely check the 9 possible man woman pairs (Mi, Wj) with $1 \leq i \leq 3$ and $1 \leq j \leq 3$. In every case the woman would not want to change (and in some cases the man also wouldn't want to change).

If input.txt contains

```
3
1 2 3
2 3 1
3 1 2
3 2 1
1 3 2
2 1 3
1 3
2 1
3 2
```

the the output should be just the number of instabilities.

```
3
```

More detailed explanation of sample input/output:
The input file describes an instance with the preference lists

M1	W1	W2	W3
M2	W2	W3	W1
M3	W3	W1	W2

W1	M3	M2	M1
W2	M1	M3	M2
W3	M2	M1	M3

and the matching $\{(M1, W3), (M2, W1), (M3, W2)\}$. Checking the 9 man/woman pairs:
M1:W1 - M1 would switch, W1 would not switch.
M1:W2 - M1 would switch, W2 would switch (INSTABILITY).
M1:W3 - already part of the matching.
M2:W1 - already part of the matching.
M2:W2 - M2 would switch, W2 would not switch.

M2:W3 - M2 would switch, W3 would switch (INSTABILITY).
M3:W1 - M3 would switch, W1 would switch (INSTABILITY).
M3:W2 - already part of the matching.
M3:W3 - M3 would switch, W3 would not switch.
So there are exactly 3 instabilities.

Stray Thoughts:
I suggest you finish you finish and submit your project at least several days in advance. This way you have time and opportunity to ask any last questions and verify that what you upload satisfies the requirements. There is nothing wrong with working on the project for a day and uploading your code, working for another day and uploading your improved code, ..., working for another day and uploading your final version. In fact there are advantages: you have a fairly reliable place that keeps your versions and even if you get busy at the last moment you have still uploaded your best version.
Your project should be written and understood by you. Helping or receiving help from others to figure out what is allowed/required is fine, but copying code is not. Significant shared source code indicates that you either did not write/understand what you submitted or you assisted another in submitting code they did not write/understand.