

## COMP 482 Project 1: PreProcessing for ProposeDispose

Due: 10/8

Points: 30 points possible

**Overview:** The STABLEMARRIAGE problem receives its input as ranked lists. Every man ranks the women from most to least desired and every woman ranks the men from most to least desirable. Recall that running the PROPOSEDISPOSE algorithm in the most obvious way using these ranked lists results in an  $O(N^3)$  algorithm where  $N$  is the number of men (or women). The reason this occurs is that when an engaged woman receives a proposal, she must linearly scan her list to determine which of the men she prefers. In class we showed a preprocessing step that could be done to reduce the time to  $O(N^2)$ . In this project you'll be given the matrix (2-d array) containing the women's rankings and need to perform this preprocessing step.

**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 value which will be the value of  $N$ . The next  $N$  lines will be the preference lists for the  $N$  women.

Your program will read in this file and perform the steps necessary to create the array where row  $i$  column  $j$  tells you how woman  $i$  ranks man  $j$ . For example if a woman ranks the men **5 1 3 8 2 7 6 4** then you should modify it to: man 1 is ranked 2nd, man 2 is ranked 5th, man 3 is ranked 3rd, man 4 is ranked 8th, man 5 is ranked 1st, man 6 is ranked 7th, man 7 is ranked 6th, and man 8 is ranked 4th. So you should create the line **2 5 3 8 1 7 6 4**.

Then you will print this array out to the screen using tabs to separate items in the same row. See the sample output below.

You can discuss the algorithm to be used with anyone and consult any source (books, internet, etc). However, for projects, you are expected to write the code on your own with limited or no assistance from the professor (using Project0.java is permitted), no assistance from others, and limited or no assistance from other sources (books, internet, etc).

**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 the correct 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.

**Sample execution:** If input.txt contains

```
4
1 2 3 4
4 3 2 1
1 3 2 4
3 1 4 2
```

then your program should output

```
1      2      3      4
4      3      2      1
1      3      2      4
2      4      1      3
```

If input.txt contains

```
6
3 2 5 4 1 6
4 3 5 2 6 1
1 6 2 4 3 5
5 1 4 6 2 3
5 1 4 6 3 2
3 2 1 6 5 4
```

then your program should output

```
5      2      1      4      3      6
6      4      2      1      3      5
1      3      5      4      6      2
2      5      6      3      1      4
2      6      5      3      1      4
3      2      1      6      5      4
```

If input.txt contains

```
8
6 1 3 2 4 5 8 7
4 3 2 1 8 7 6 5
1 3 5 7 2 4 6 8
1 8 2 7 3 6 4 5
4 5 3 6 2 7 1 8
2 4 6 8 1 3 5 7
8 7 6 5 4 3 2 1
5 1 3 8 2 7 6 4
```

then the output should be

```
2      4      3      5      6      1      8      7
4      3      2      1      8      7      6      5
1      5      2      6      3      7      4      8
1      3      5      7      8      6      4      2
7      5      3      1      2      4      6      8
5      1      6      2      7      3      8      4
8      7      6      5      4      3      2      1
2      5      3      8      1      7      6      4
```

**Stray Thoughts:**

I will be using a recent version of Java (likely the current version Java SE 17, but if a new version is released I may upgrade).

You are generally allowed to use the standard features, classes, methods in Java. For example, I expect nearly all students will sometimes want to use either an array or `java.util.ArrayList` and the built in sort routine (either for arrays or `ArrayLists`). This is allowed as long as it doesn't violate a project requirement. You can use as many or as few files as you feel appropriate, but the main method should be located in a file called `Project1.java`. Otherwise the project won't compile/run with the required commands.

Some IDEs default to placing java files into packages. This will likely cause the commands '`javac Project1.java`' and/or '`java Project1`' to fail. Either use an IDE that does not place java files into packages OR learn your preferred IDE well enough to avoid this issue OR delete any package lines before submission.

Students often decide to change or modify the format of the input or output. Sometimes it makes the project easier for them. Other times a student thinks it is an improved design. You may or may not be right, but don't change the input or output format.

It is likely that many students won't read this far, but there is no need to let me know you've read this.

I suggest you finish your project several days in advance. This way you have time and opportunity to ask any last questions, you don't get caught by a power outage or similar issue, and verify that what you upload satisfies the requirements.

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 or understand.

These project description files are getting excessively long, but I find that failing to state things which are common sense yields complaints.