Question 1a

Names:

Tristan Cheah, Owen Quek

**Requirements/Specifications**

This program showcases the Gale-Shapely Algorithm to output a set of stable matches.

The input is done in the format as provided in the assignment brief. All inputs must have all interactable entities listed properly. For example, if Hospital C and M exist, the preference list of the Staff must include both C and M.

**User Guide**

Run a WSL(Windows Subsystem Linux) Window.

Navigate to the appropriate directory

Type /make to build the program

Type ./algoq1.out <name of text input> (example: ./algoq1.out text.txt)

Output will be in the WSL Window

**Design and Analysis**

The design of this program was based on the pseudocode of the Gale Shapley algorithm provided in the Week 1 presentation about stable matching. For Question 1a, we had to take in a text-based input so we use hard-coded data structures at first to decide how we wanted to read the data before doing the text-based input.

For question 1b, I adapted the existing algorithm written in question 1a to fit the requirements of the question. Namely the preference list must be generated based on the requirements of the question

**Limitations**

All inputs must have all interactable entities listed properly. For example, if Hospital C and M exist, the preference list of the Staff have to include both C and M.

**Testing**

**Question 1a**

The following inputs were tested

Input:

C: < 5, 1, 2, 4, 3 >; M: < 5, 3, 1, 2, 4 >;

1: < C, M >; 2: < C, M >; 3: < C, M >;.

Output:

{ 1, C }

{ 2, C }

{ 3, M }

Input:

C: < 5, 1, 2, 4, 3 >; M: < 5, 3, 1, 2, 4 >;

1: < C, M >; 2: < C, M >; 3: < C, M >; 4: < M, C >; 5: < M, C >.

Output:

{ 1, C }

{ 2, C }

{ 3, }

{ 4, M }

{ 5, M }

Input:

C: < 3, 1, 2, 5, 4 >; M: < 5, 3, 1, 2, 4 >;

1: < C, M >; 2: < C, M >; 3: < C, M >; 4: < M, C >; 5: < M, C >.

Output:

{ 1, }

{ 2, C }

{ 3, C }

{ 4, M }

{ 5, M }

**Question 1b**

With hardcoded data set from assignment brief

{ Staff ID: 7, Assigned To: 4 }

{ Staff ID: 8, Assigned To: 4 }

{ Staff ID: 3, Assigned To: 3 }

{ Staff ID: 2, Assigned To: 3 }

{ Staff ID: 1, Assigned To: 2 }

{ Staff ID: 6, Assigned To: 2 }

{ Staff ID: 4, Assigned To: 1 }

{ Staff ID: 5, Assigned To: 1 }

**Listings**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**Question 1b**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**