Stable Marriage
Problem ID: 1marriage

Introduction

”Life’s a game made for everyone. And love is the prize.” (Wake me up – Avicii)

Everywhere we look, love is featured. In movies, series, commercials, when you go for a walk and some couple are kissing (annoyingly) passionatly on your favourite park bench. Furthermore various articles and books are written on the subject, suggesting that it is of great interest to many people.

Finding love is a tough challenge for everyone. Sometimes you like someone who does not love you. Sometimes someone likes you who you don’t like. Sometimes you like each other, but then a third person appears and causes a love triangle.

Aims

The goal of the lab is:

- Implementing an algorithm.
- Debugging your code.
- Structuring your code in a logical fashion.
- Parsing messy input.
- Reason about correctness of your algorithm.
- Reason about upper bounds for time complexity.

Problem formulation

Please note that the algorithm was invented (by David Gale and Lloyd Shapley) and the problem was formulated in 1962[1] with binary, heterosexual, monogamous relationships. To ease the understanding of the problem we keep the formulation, but the student is encuraged to change ”men” and ”women” to some binary classes of e.g. hospitals and medical students or servers and clients.

You are given each man’s and woman’s preference lists. Given these lists find a stable matching, meaning that there do not exist two man-woman-pairs $(m_1, w_1), (m_2, w_2)$ such that w_2 is higher ranked on m_1’s preference list than w_1 and simultaneously m_1 is higher ranked on w_2’s preference list than m_2 (thus that both m_1 and w_2 would prefer to change partner).

Input

Input is given through standard in (not from a file). To run the program write for instance

- java solution_file < input_file
- python3 solution_file < input_file
- ./a.out < input_file

The first line of the input contains a single integer N ($1 \leq N \leq 3000$), the number of men and women.

Then follows the preference lists. In total $(N + 1) \cdot 2N$ integers are given on one or more lines. These numbers are structured in $2N$ chunks of $N + 1$ integers each. Each chunk contain the description of one persons preference list – first an integer $1 \leq i \leq N$, the index of the person, then N distict integers, the preference list of person i. These N integers are all different and between 1 and N, inclusive.

Note that each person index $i \in \{1, 2, \ldots, N\}$ appears twice – the first appearence is for woman i, the second for man i.

[1] https://en.wikipedia.org/wiki/Stable_marriage_problem#Algorithm

Output

The output should be printed to standard out (not to a file). To do this use for instance

- System.out.println() in Java
- print() in Python
- cout in C++.

The output should consist of exactly N rows. Row i should contain exactly one integer, the index of the man paired with woman i.

Examination and Points of Discussion

To pass the lab make sure you:

- Have successfully implemented the algorithm with time complexity $\mathcal{O}(n^2)$.
- Have neat and understandable code.
- Have descriptive variable names.
- Have filled in the blanks in the report.
- Have run the check_solution script, to validate your solution.

During the oral presentation you will discuss the following questions with the lab assistant:

- Why does your algorithm obtain a stable solution?
- Could there be other stable solutions? Do you have an example/proof of uniqueness?
- What is the time complexity and why?
- Is this (the algorithm) how matching is performed in real life? If not, what flaws does it have?
- Are there any applications of the algorithm (as it is or in a slightly shifted version)?

Sample Input 1

```
2
1 1 2
2 2 1
1 1 2
2 2 1
```

Sample Output 1

```
1
2
```

Sample Input 2

```
2
1 1 2
1 2 1
2 2 1
2 1 2
```

Sample Output 2

```
2
1
```