

Programming Assignment 1

- Due date: Thursday March 31st at 11.59pm on Gradescope (link on Canvas)
- The programming assignments are mandatory for students enrolled in 5350. If you are enrolled in 4350, you are allowed to work on these too. For 4350 students I will only take your completion of these exercises into account when you are a borderline case between two letter grades.
- You can use Python or java for this programming exercise. If you want to use a different language, make sure to ask on Ed — I will probably say yes.
- The programming assignments may be completed in groups of 2, and groups may change between assignments. There will be a peer evaluation at the end of the semester so make sure that you contribute sufficiently!

Set up

These programming exercises will be for 2-player static games. The input will be from the standard input, and you should show your results using standard output as well.

The input consists of the following:

- the first line of the input specifies the number of strategies for player 1
- the second line contains the names of the strategies of player 1, separated by white space
- the third line specifies the number of strategies for player 2
- the fourth line contains the names of the strategies of player 2, separated by white space
- the next lines specify the payoff as a bimatrix:
 - the first of these lines correspond to the first strategy for player 1, and will give the payoffs for player 1 and player 2 for all of the strategies of player 2
 - the payoffs of player 1 and player 2 are separated by a comma
 - the payoffs for the different strategies for player 1 and 2 are separated by white space.

Implement the payoffs as doubles.

Example input

Here is an example input for the game Matching Pennies, where player 2 has the additional strategy S (stupid).

```
2
H T
3
H T S
1,-1 -1,1 0,-100
-1,1 1,-1 0,-100
```

In addition to the above, the input will have 0–2 additional lines, depending on what we want you to compute. The details on these lines of the input, as well as the required output formatting, are given below.

Exercises

1. Make a program that finds dominated strategies for a specific player.

After the input of the game, there will be one more line with a player number (1 or 2). Your program should output as many lines as there are pairs of strategies, one dominating the other.

For instance, if we want to find the dominated strategies for player 2, we add the following line to the input:

```
2
```

and the expected output is

```
S H
S T
```

meaning that S is dominated by both H and T.

2. Make a program that finds the set of best responses.

After the input of the game, there will be two more lines: one with a player number (1 or 2) and the second with a strategy from her strategy set. For example:

```
1
H
```

Your program should output 1 line with all strategies that are best responses to the strategy specified on the last line of the input for the player specified on the penultimate input line, separated by white space.

For the example:

```
T
```

3. Make a program that finds the set of IESDS strategies (in pure-strategies).

There is no additional input. The output are lines that contain pairs of strategies, first player 1's strategy, then player 2's, that are IESDS strategies.

For the example:

```
H H
H T
T H
T T
```

4. Make a program that finds the set of rationalizable strategies (in pure-strategies).

There is no additional input. The output are lines that contain pairs of strategies, first player 1's strategy, then player 2's, that are rationalizable.

5. Make a program that finds all pure-strategy Nash equilibria.

There is no additional input. The output are lines that contain pairs of strategies, first player 1's strategy, then player 2's, that are pure-strategy Nash equilibria.

6. Make a program that finds all mixed-strategy Nash equilibria.

There is no additional input. The output are lines that contain mixed-strategies: the first line contains the mixed-strategy for player 1, given as a number (the probability) for each pure-strategy. The second line contains the mixed-strategy for player 2. If there are more than one Nash equilibria, this is repeated as often as necessary.

For the example, the expected output would be:

```
0.5 0.5
0.5 0.5 0
```

For this last question, you will get full credit if it works correctly for games where players have up to 3 strategies each.