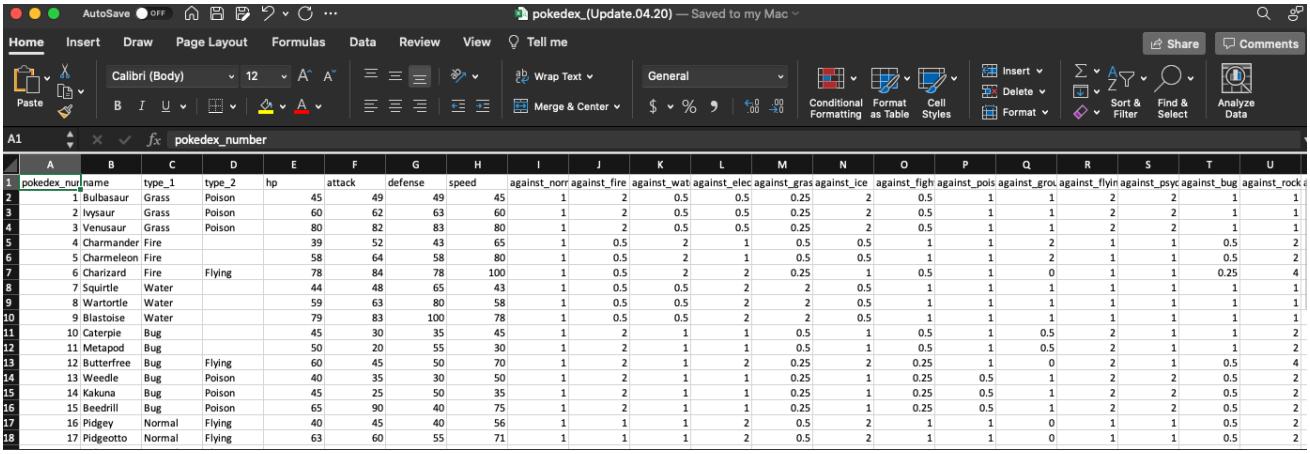
CMPS 280

Description

For this assignment you will create a Pokedex program. Your program will read the data from a provided file (`pokedex.csv`), and will as the user to input the number of two Pokemon. The program will then tell which pokemon has an advantage in battle.

The advantage will be given in the `pokedex.csv` file. Don't worry about the `.csv` extension, all it is is a text file formatted in a specific way.

You can open the `pokedex.csv` with Microsoft Excel, and it will look something like this:



pokedex_nur	name	type_1	type_2	hp	attack	defense	speed	against_norr	against_fire	against_wat	against_elec	against_gras	against_ice	against_figh	against_pois	against_gro	against_flyin	against_psyc	against_bug	against_rock
1	Bulbasaur	Grass	Poison	45	49	49	45	1	2	0.5	0.5	0.25	2	0.5	1	1	2	2	1	1
2	Ivysaur	Grass	Poison	60	62	63	60	1	2	0.5	0.5	0.25	2	0.5	1	1	2	2	1	1
3	Venusaur	Grass	Poison	80	82	83	80	1	2	0.5	0.5	0.25	2	0.5	1	1	2	2	1	1
4	Charmander	Fire		39	52	43	65	1	0.5	2	1	0.5	0.5	1	1	2	1	1	0.5	2
5	Charmeleon	Fire		58	64	58	80	1	0.5	2	1	0.5	0.5	1	1	2	1	1	0.5	2
6	Charizard	Fire	Flying	78	84	78	100	1	0.5	2	2	0.25	1	0.5	1	0	1	1	0.25	4
7	Squirtle	Water		44	48	65	43	1	0.5	0.5	2	2	0.5	1	1	1	1	1	1	1
8	Wartortle	Water		59	63	80	58	1	0.5	0.5	2	2	0.5	1	1	1	1	1	1	1
9	Blastoise	Water		79	83	100	78	1	0.5	0.5	2	2	0.5	1	1	1	1	1	1	1
10	Caterpie	Bug		45	30	35	45	1	2	1	1	0.5	1	0.5	1	0.5	2	1	1	2
11	Metapod	Bug		50	20	55	30	1	2	1	1	0.5	1	0.5	1	0.5	2	1	1	2
12	Butterfree	Bug	Flying	60	45	50	70	1	2	1	2	0.25	2	0.25	1	0	2	1	0.5	4
13	Weedle	Bug	Poison	40	35	30	50	1	2	1	1	0.25	1	0.25	0.5	1	2	2	0.5	2
14	Kakuna	Bug	Poison	45	25	50	35	1	2	1	1	0.25	1	0.25	0.5	1	2	2	0.5	2
15	Beedrill	Bug	Poison	65	90	40	75	1	2	1	1	0.25	1	0.25	0.5	1	2	2	0.5	2
16	Pidgey	Normal	Flying	40	45	40	56	1	1	1	2	0.5	2	1	1	0	1	1	0.5	2
17	Pidgeotto	Normal	Flying	63	60	55	71	1	1	1	2	0.5	2	1	1	0	1	1	0.5	2

Behind the scenes, all it is is a text file, in which each line has the data of a single pokemon. The data for a single pokemon is then separated by commas (`,`). Here are the first few lines of the pokedex (minus the first line):

```
1,Bulbasaur,Grass,Poison,45,49,49,45,1,2,0.5,0.5,0.25,2,0.5,1,1,2,2,1,1,1,1
2,Ivysaur,Grass,Poison,60,62,63,60,1,2,0.5,0.5,0.25,2,0.5,1,1,2,2,1,1,1,1,0
3,Venusaur,Grass,Poison,80,82,83,80,1,2,0.5,0.5,0.25,2,0.5,1,1,2,2,1,1,1,1,0
4,Charmander,Fire,,39,52,43,65,1,0.5,2,1,0.5,0.5,1,1,2,1,1,0.5,2,1,1,0.5
5,Charmeleon,Fire,,58,64,58,80,1,0.5,2,1,0.5,0.5,1,1,2,1,1,0.5,2,1,1,0.5
6,Charizard,Fire,Flying,78,84,78,100,1,0.5,2,2,0.25,1,0.5,1,0,1,1,0.25,4,1,1
```

Notice that some Pokemons only have a single type, and therefore the column for `type_2` is empty (see `Charmander` for example). The first line contains the names of the attributes of the pokemon, it is called the **header**, here is a description of each column:

- `pokedex_number`: The entry number of the Pokemon in the National Pokedex
- `name`: The English name of the Pokemon
- `type_1`: The Primary Type of the Pokemon
- `type_2`: The Secondary Type of the Pokemon if it has it

- `hp`: The Base HP of the Pokemon
- `attack`: The Base Attack of the Pokemon
- `defense`: The Base Defense of the Pokemon
- `speed`: The Base Speed of the Pokemon
- `against_normal`: Denotes the multiplier applied when damage is taken from an attack of a normal type pokemon
- `against_fire`: Denotes the multiplier applied when damage is taken from an attack of a fire type pokemon
- `against_water`: Denotes the multiplier applied when damage is taken from an attack of a water type pokemon
- `against_electric`: Denotes the multiplier applied when damage is taken from an attack of an electric type pokemon
- `against_grass`: Denotes the multiplier applied when damage is taken from an attack of a grass type pokemon
- `against_ice`: Denotes the multiplier applied when damage is taken from an attack of a ice type pokemon
- `against_fight`: Denotes the multiplier applied when damage is taken from an attack of a fighting type pokemon
- `against_poison`: Denotes the multiplier applied when damage is taken from an attack of a poison type pokemon
- `against_ground`: Denotes the multiplier applied when damage is taken from an attack of a ground type pokemon
- `against_flying`: Denotes the multiplier applied when damage is taken from an attack of a flying type pokemon
- `against_psychic`: Denotes the multiplier applied when damage is taken from an attack of a psychic type pokemon
- `against_bug`: Denotes the multiplier applied when damage is taken from an attack of a bug type pokemon
- `against_rock`: Denotes the multiplier applied when damage is taken from an attack of a rock type pokemon
- `against_ghost`: Denotes the multiplier applied when damage is taken from an attack of a ghost type pokemon
- `against_dragon`: Denotes the multiplier applied when damage is taken from an attack of a dragon type pokemon
- `against_fairy`: Denotes the multiplier applied when damage is taken from an attack of a fairy type pokemon

This dataset is adapted from a more comprehensive one found at: https://www.kaggle.com/mariotormo/complete-pokemon-dataset-updated-090420 The file format `.csv` stands for `Comma Separated Values`, you can learn more about it here: https://en.wikipedia.org/wiki/Comma-separated_values

Requirements

- Your program must have at least 3 different classes designed and implemented by yourself: `Pokemon` and `Pokedex`, and `Main`
- The class `Pokemon` should be used to represent a single pokemon
- The `Pokedex` class must implement the `public Pokemon typeAdvantageCheck(Pokemon p1, Pokemon p2);` method, that returns the pokemon that has the advantage against the other.
- The `Main` class is where the `public static void main(String[] args) {}` method is located, where you will implement the main logic of your program
- At least one of the classes (`Pokemon` or `Pokedex`) must `throw` an exceprion (can be a custom made exception or a ready-to-use provided by Java)
- Your program must use the `try/catch` block at least once
- Your submission will be a compressed file containing all the `.java` source files (do **not** include `.class` files or the dataset)
- Your program must compile!

Hints

- Investigate the most suitable way to read the contents from the dataset (official documentation, FileReader and the class material are a good start)
- **Test, test, test, and test often!** Run your program after making significant changes to your code, and make sure to fix any issue that shows up while testing
- Sharing advices and helping each other is encouraged
- Copying from a colleague is **prohibited** and **will** grant you a zero grade
- Being on the right track can still give you marks, even if the program has bugs
- Be creative and **have fun**!

Example run

```
Enter the first pokemon number (1–151): 4
You chose Charmander.
Enter the second pokemon number (1–151): 7
You chose Squirtle.
Squirtle has the type advantage over Charmander in a battle!
```