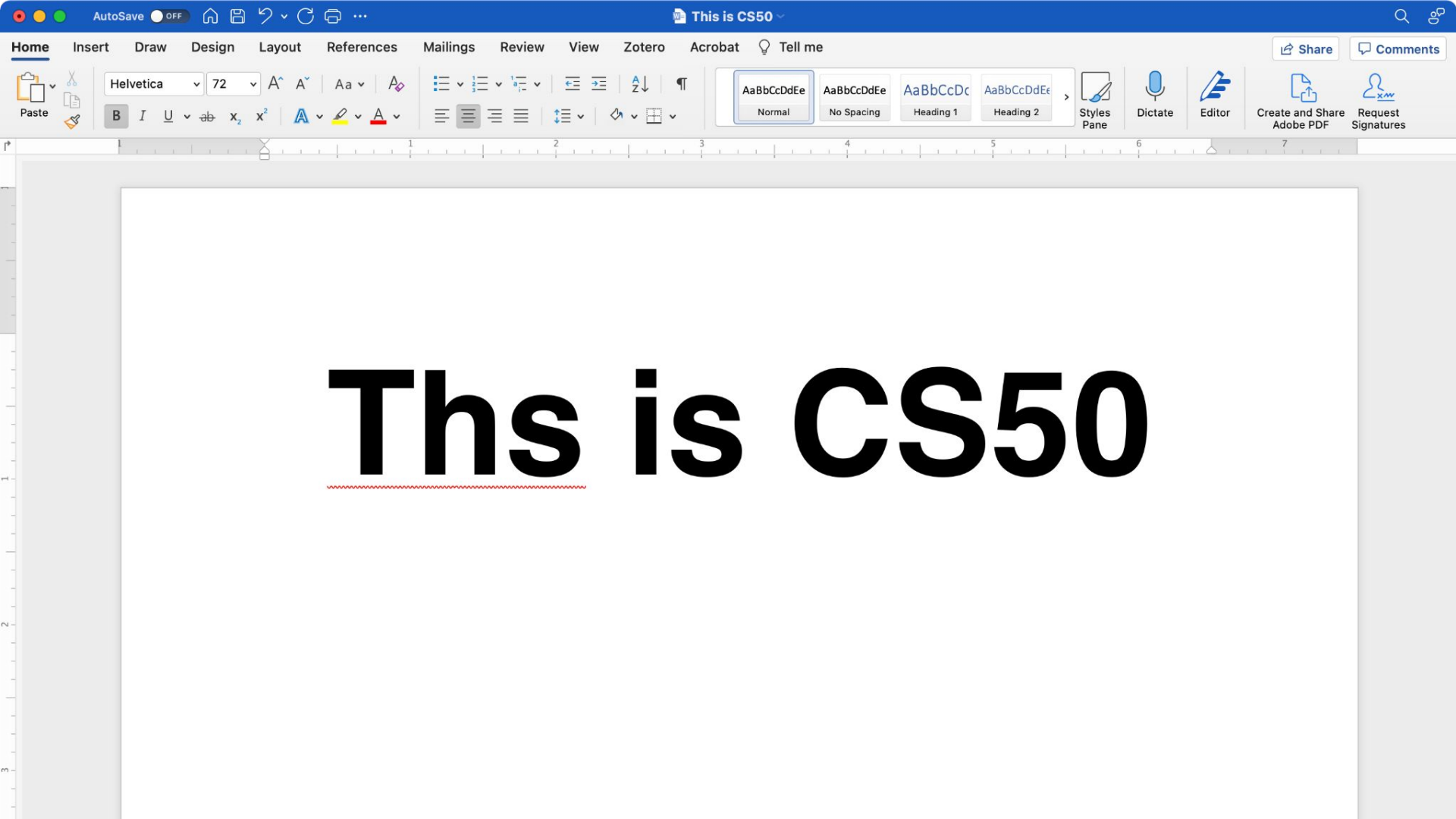

AutoSave OFF
This is CS50
Home Insert Draw Design Layout References Mailings Review View Zotero Acrobat Tell me
Share Comments
Paste
Helvetica 72
AaBbCcDdEe Normal
AaBbCcDdEe No Spacing
AaBbCcDc Heading 1
AaBbCcDdEe Heading 2
Styles Pane
Dictate
Editor
Create and Share Adobe PDF
Request Signatures
Ths is CS50

Think.
Pair.
Share.

cs50.ly/questions

```
string phrase = get_string("");
```

```
phrase = input("")
```

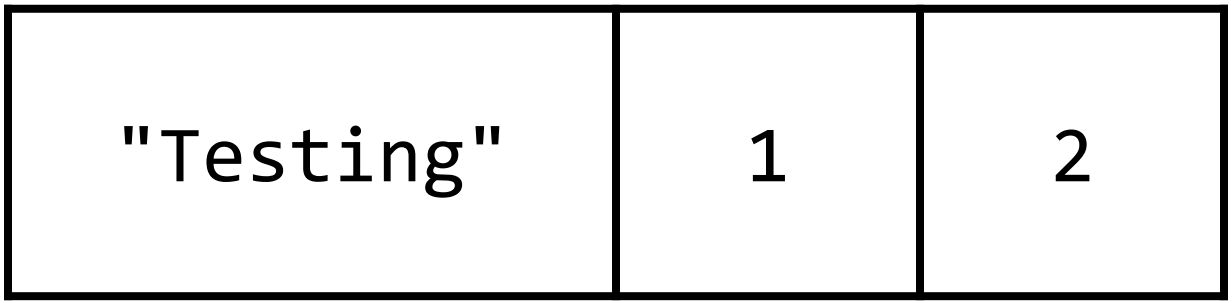
```c
if (strcmp(phrase, "hello") == 0)
{
    printf("Hi, %s!\n", name);
}
```

```python
if phrase == "hello":
    print(f"Hi {name}!")
```

```
my_list = ["Testing", 1, 2]
```

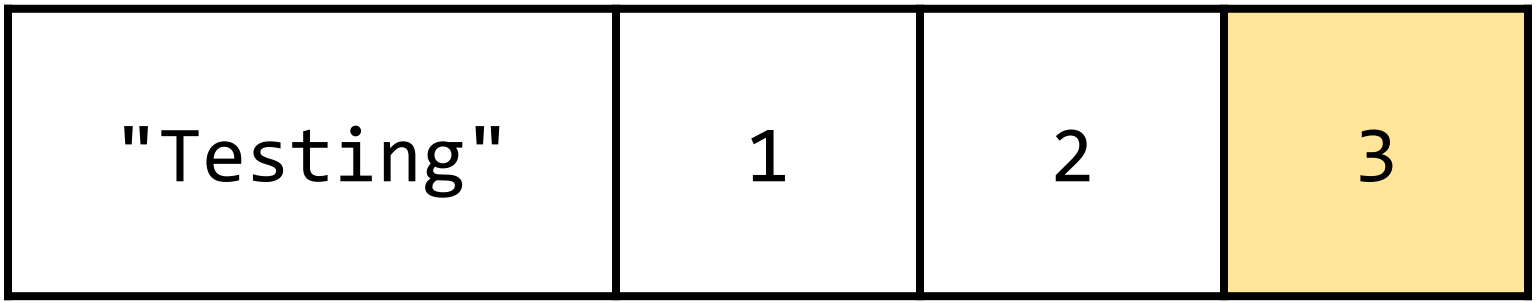
```
my_list = ["Testing", 1, 2]
```

my_list



```
my_list.append(3)
```

my_list



```
for i in [0, 1, 2]:
    print(i)
```

```
for i in range(0, 3, 1):
    print(i)
```

Start (inclusive)

```
for i in range(0, 3, 1):
    print(i)
```

End (exclusive)

```
for i in range(0, 3, 1):
    print(i)
```

Step

```
for i in range(0, 3, 1):
    print(i)
```

```
phrase[i]
```


```
phrase[i]
```

String Predictions

Download and open **str_prediction.py** in code.cs50.io.

At cs50.ly/str_prediction, predict the outcomes for each "Round" of string manipulation in Python. Write your predictions **without** running **str_prediction.py**. Then, run **python str_prediction.py** to see what's actually happening!

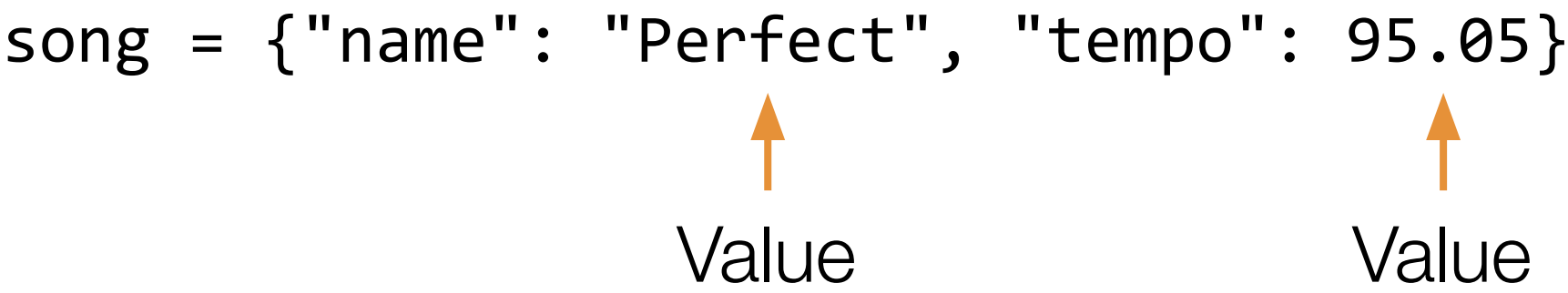
Afterwards, try modifying the **range** function in Round 6 to print a string backwards.

```
song = {"name": "Perfect", "tempo": 95.05}
```

```
song = {"name": "Perfect", "tempo": 95.05}
```

Key

Key


song = {"name": "Perfect", "tempo": 95.05}
Value
Value

```
song = {"name": "Perfect", "tempo": 95.05}
```

song	
"name"	"Perfect"
"tempo"	95.05

```
song["name"]
```

song	
"name"	"Perfect"
"tempo"	95.05

```
song["album"] = "Divide"
```

song	
"name"	"Perfect"
"tempo"	95.05
"album"	"Divide"

```
[{name: "Perfect", tempo: 95.05},
 {name: "Eastside", tempo: 89.391},
 {name: "Wolves", tempo: 124.946},
 {name: "Him & I", tempo: 87.908}]
```

```
[{name: "Perfect", tempo: 95.05},
 {name: "Eastside", tempo: 89.391},
 {name: "Wolves", tempo: 124.946},
 {name: "Him & I", tempo: 87.908}]
```

```
[{name: "Perfect", tempo: 95.05},
 {name: "Eastside", tempo: 89.391},
 {name: "Wolves", tempo: 124.946},
 {name: "Him & I", tempo: 87.908}]
```

2018 top 100

```
name,tempo
God's Plan,77.169
SAD!,75.023
rockstar (feat. 21 Savage),159.847
Psycho (feat. Ty Dolla $ign),140.124
In My Feelings,91.03
Better Now,145.028
...
```

```
with open(FILENAME) as file:
```

```
with open(FILENAME) as file:
    file_reader = csv.DictReader(file)
```

```
with open(FILENAME) as file:
    file_reader = csv.DictReader(file)
    for row in file_reader:
```

```
with open(FILENAME) as file:
    file_reader = csv.DictReader(file)
    for row in file_reader:
        ...
```

Crafting Playlists

Download **playlist.py** and **2018_top100.csv**.

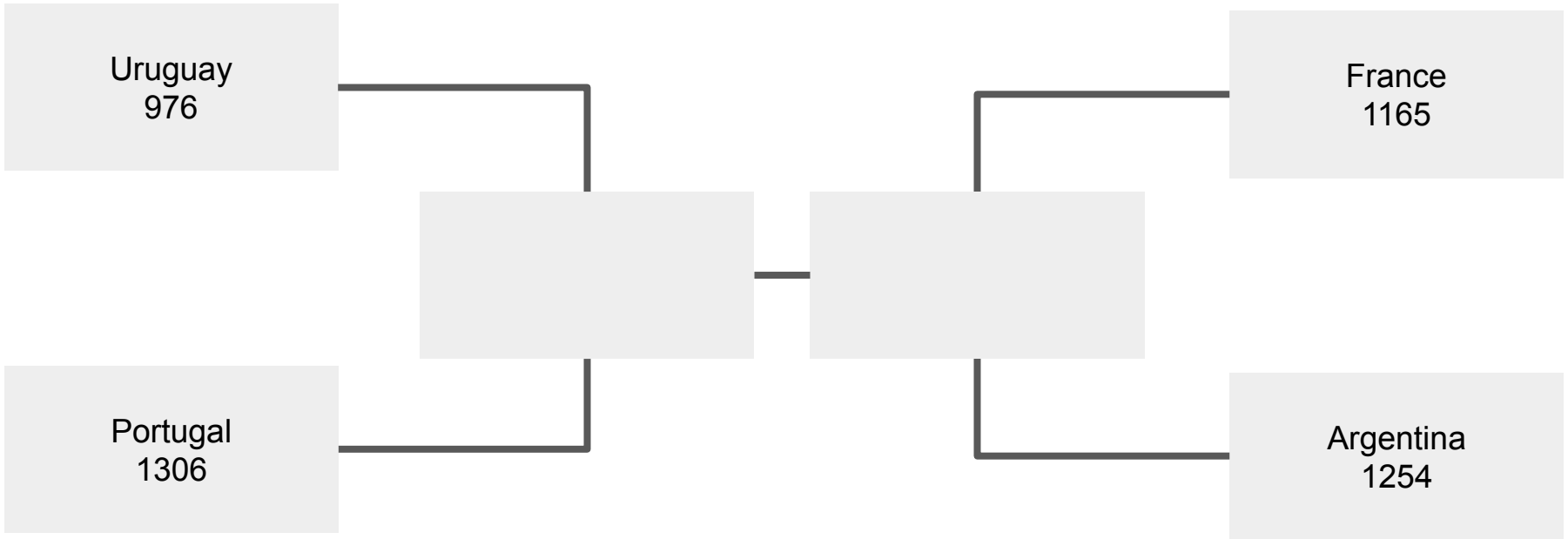
Complete the TODOs in **playlist.py** so that a user can build a playlist of popular songs within a tempo range.

If feeling more comfortable, try allowing the user to eliminate songs with certain words or phrases in the title (e.g., "love").

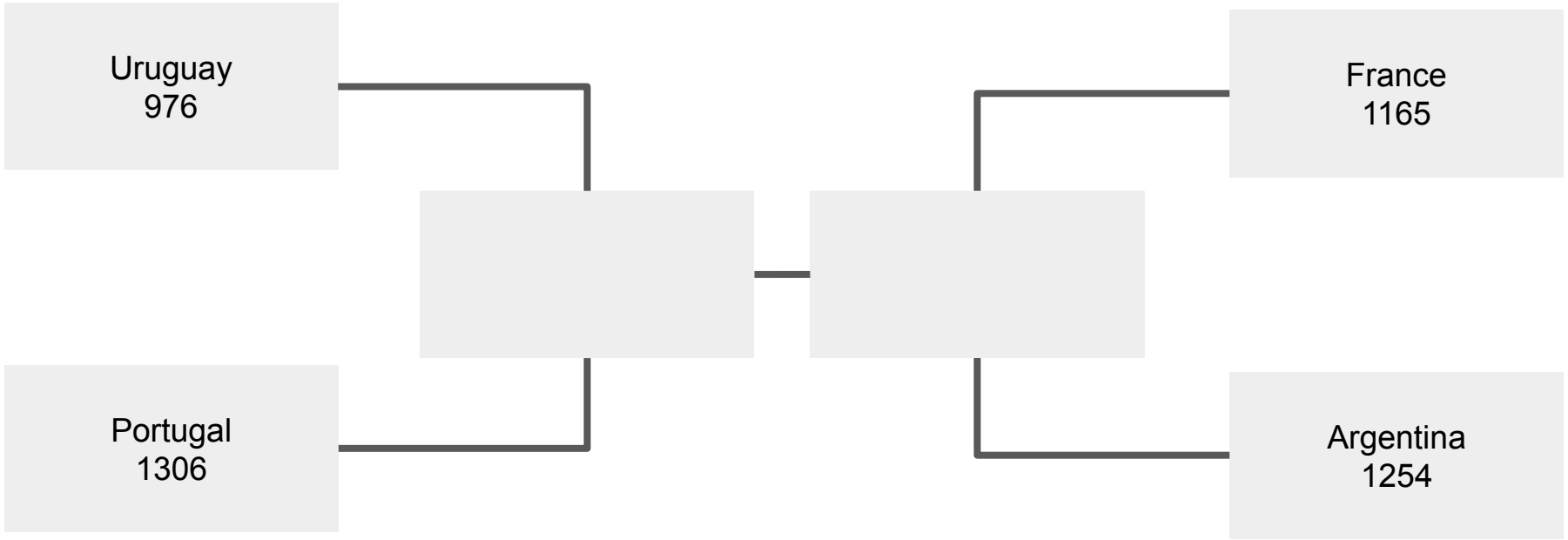
Lab

```
$ python tournament.py 2018m.csv
Belgium: 20.9% chance of winning
Brazil: 20.3% chance of winning
Portugal: 14.5% chance of winning
Spain: 13.6% chance of winning
Switzerland: 10.5% chance of winning
Argentina: 6.5% chance of winning
England: 3.7% chance of winning
France: 3.3% chance of winning
Denmark: 2.2% chance of winning
Croatia: 2.0% chance of winning
Colombia: 1.8% chance of winning
Sweden: 0.5% chance of winning
Uruguay: 0.1% chance of winning
Mexico: 0.1% chance of winning
```

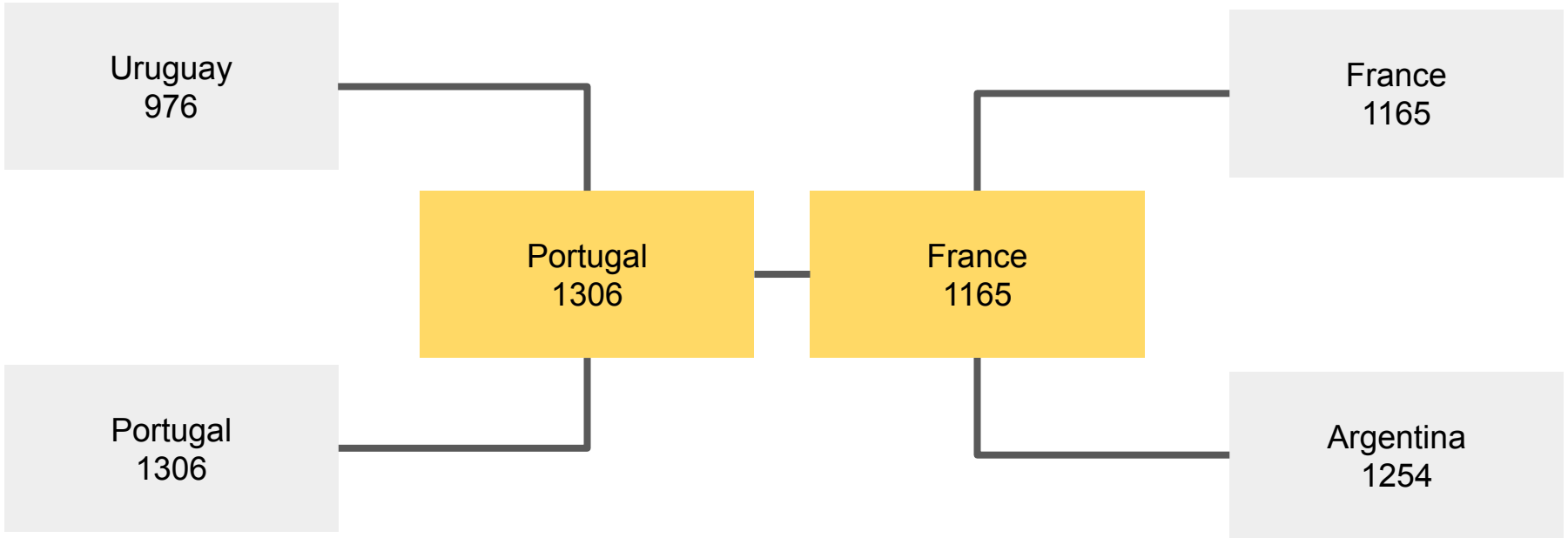
```
[{name: "Uruguay", rating: 976},
 {name: "Portugal", rating: 1306},
 {name: "France", rating: 1166},
 {name: "Argentina", rating: 1254}]
```


Uruguay
976
Portugal
1306
France
1165
Argentina
1254

simulate_round(teams)

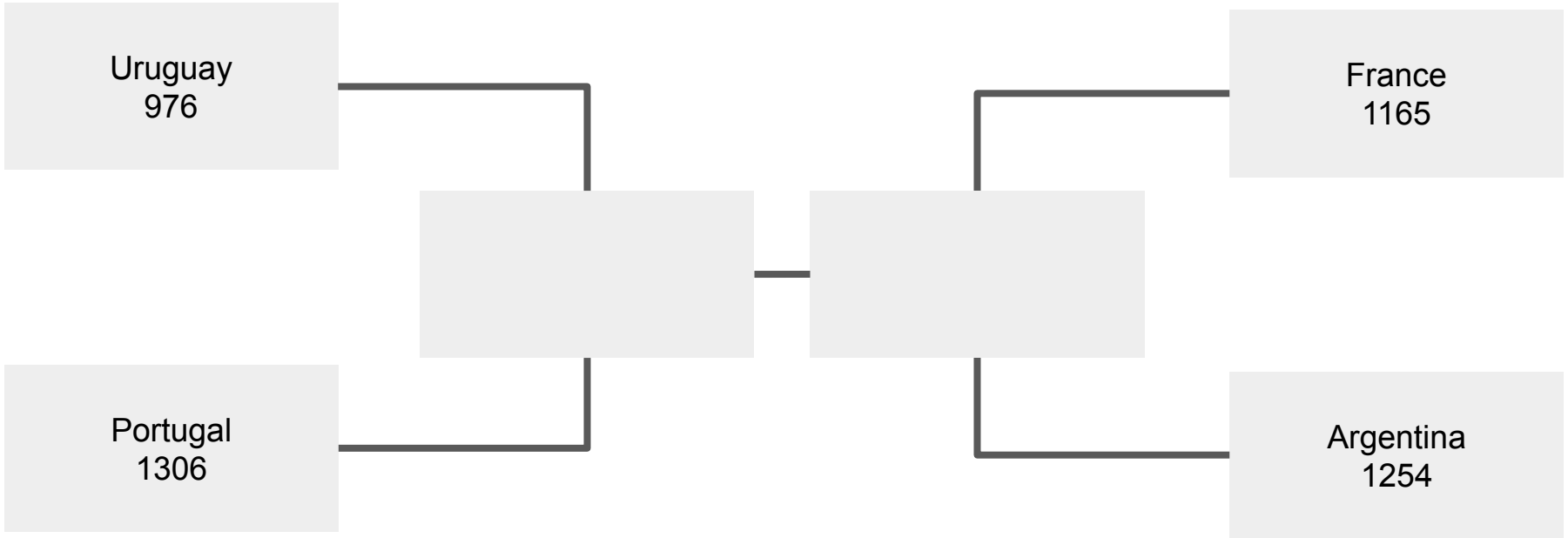


simulate_round(teams)

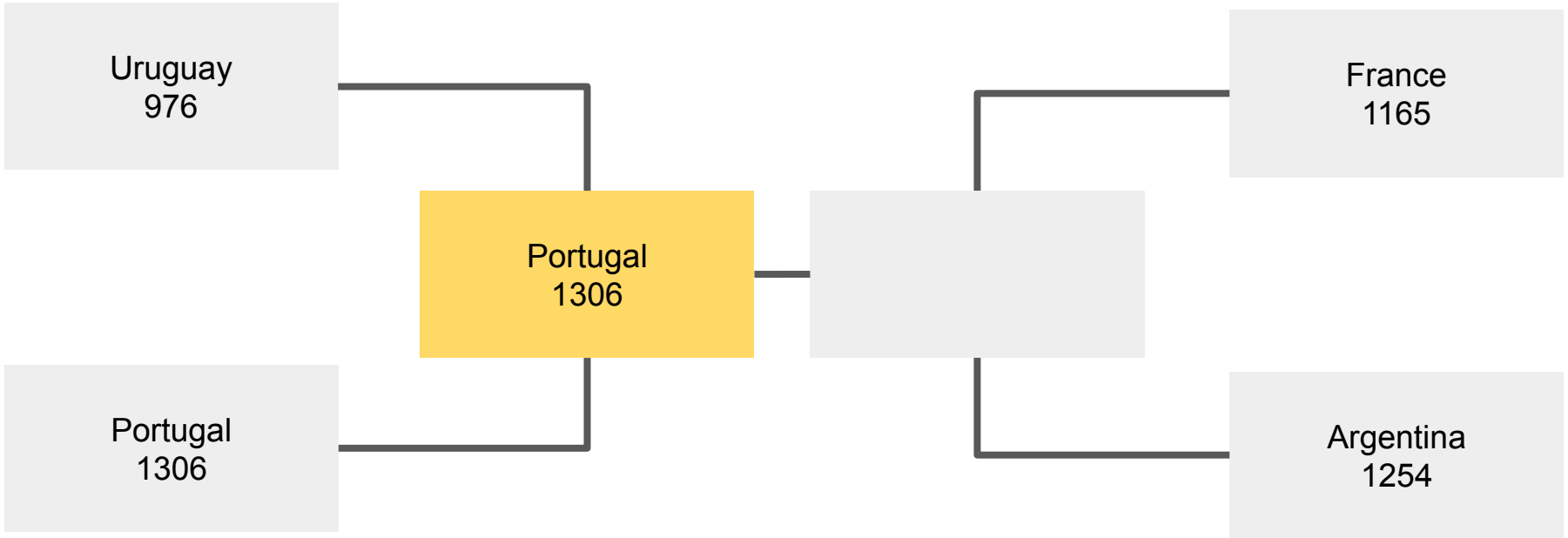


```
[{name: "Portugal", rating: 1306},
 {name: "France", rating: 1166}]
```

simulate_game(team1, team2)

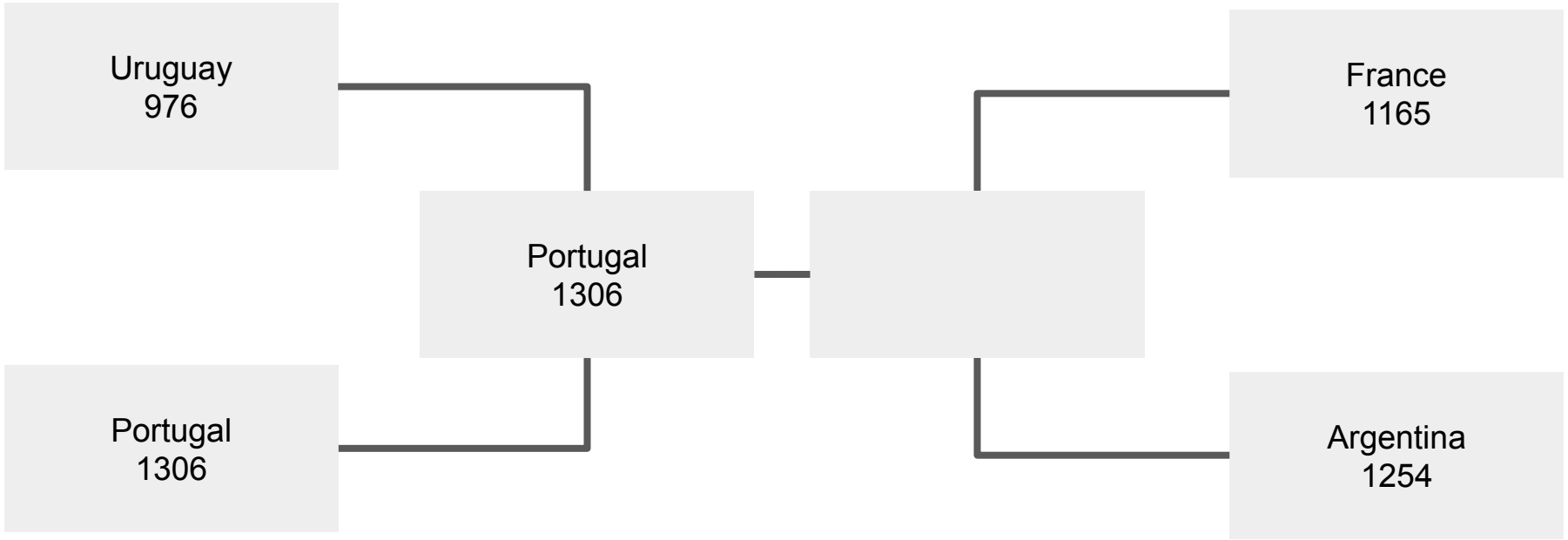


simulate_game(team1, team2)

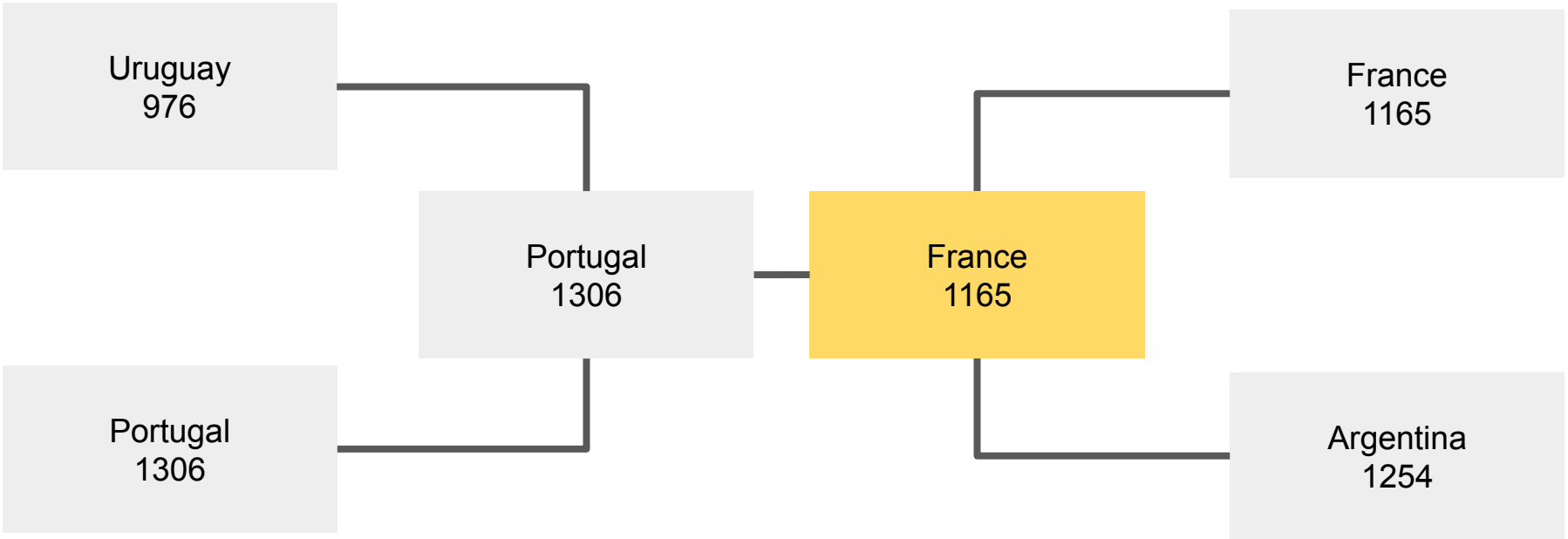


```
{name: "Portugal", rating: 1306}
```

simulate_game(team1, team2)

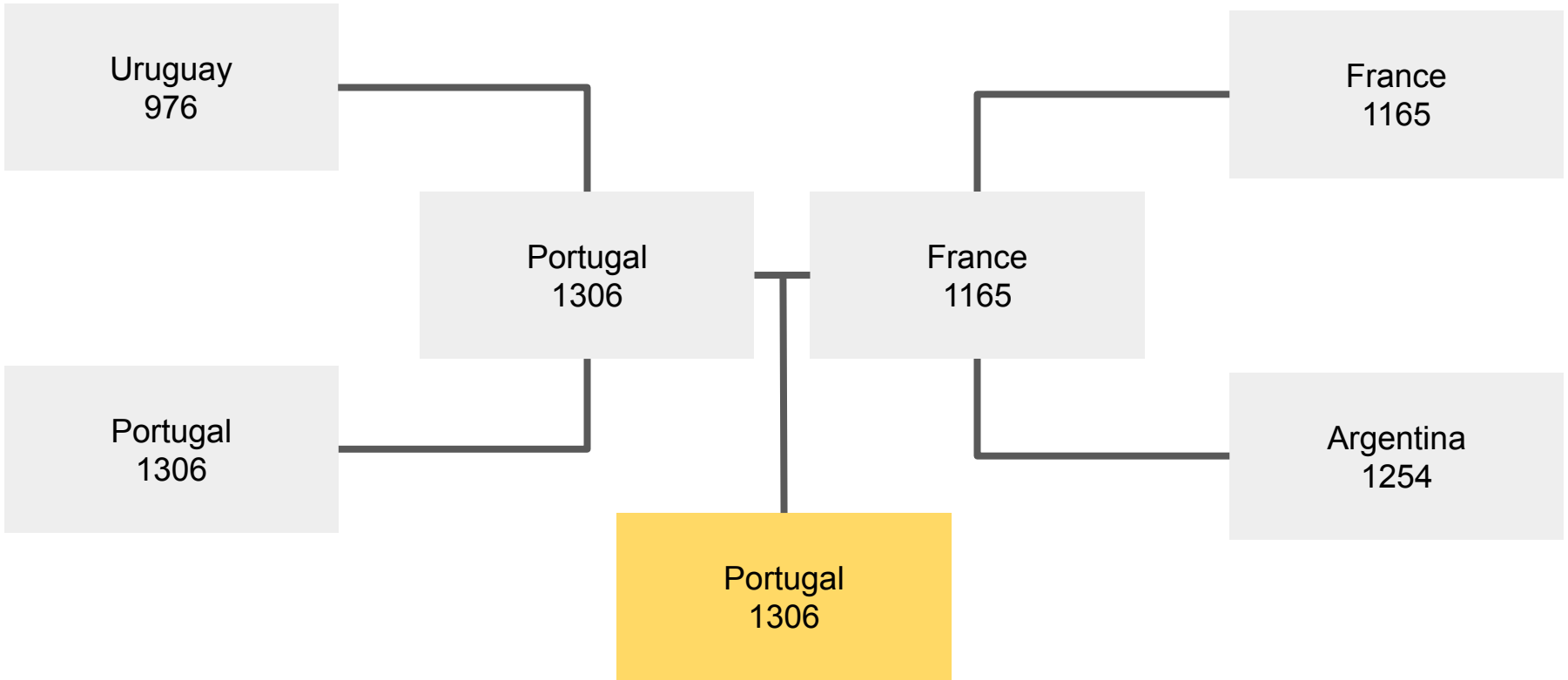


simulate_game(team1, team2)



```
{name: "France", rating: 1165}
```

simulate_round(teams)



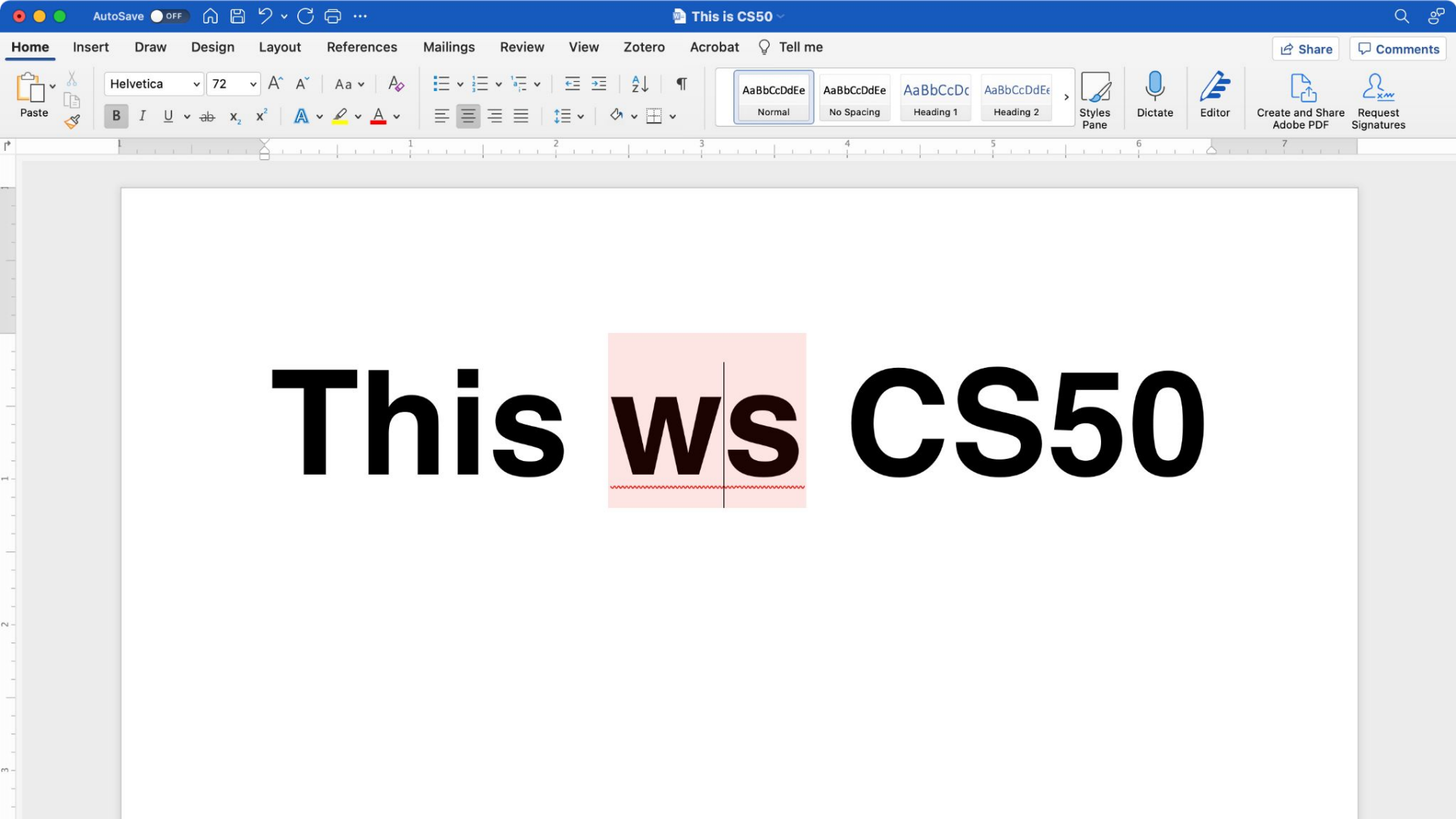
TODOs

1. Load teams from CSV into a list, with each team represented as a dictionary.
2. Complete **simulate_tournament**, using **simulate_round**, returning the name of a tournament winner.
3. Run **simulate_tournament** N times, keeping track of how many simulations a team wins.

Tutorials

Office Hours

cs50.ly/attend


AutoSave OFF
This is CS50
Home Insert Draw Design Layout References Mailings Review View Zotero Acrobat Tell me
Share Comments
Paste
Helvetica 72
AaBbCcDdEe Normal
AaBbCcDdEe No Spacing
AaBbCcDc Heading 1
AaBbCcDdEe Heading 2
Styles Pane
Dictate
Editor
Create and Share Adobe PDF
Request Signatures
This ws CS50