BVA 501E Assignment 1

Due 12 November 2023

Submission Guidelines:

- Please submit a single python code file (your submission should have .py extension). You can use the template given to you with this assignment.
- You can work on your assignment and submit it through Ninova until the midnight of the due date.
- Any questions regarding the assignment and late submissions can be directed to mergun@itu.edu.tr.
- In case you cannot submit until midnight, you can submit you homework through email between 00:00 - 00:15 on November 13,2023. Any email submissions before or after this period will be ignored.
- Please put "[BVA 501E] Assignment 1 Submission" (without quotes) to the subject of your email.
- This assignment is meant to be completed **individually**. Please **do not** share your codes with anyone for any reason.
- Enter your full name and student number into the relevant functions comes with the code template as directed.
- **All codes you submit must be in the designated function**. Writing code outside of the function scope will create errors when your file is imported.

Your task:

A GSM operator Wadafone launches a competition among its users. Each user bid on a brand new MyPhone and the user that submits the lowest unique bid will get the phone at the price he bids. Each customer can only bid exactly once. **Write a function** finds and returns **the name of the winner** and **bidding price** in a given list. If there is no unique bid, return ”No winner” for name and -1. **Exclude** bids that involves **negative price** quote.

Additional Files:

1. **assignment1_template.py**: Write your code using this template. **DO NOT** change the function names or add new parameters to the functions. You may add extra functions if needed.
2. **assignment1_test.py**:Run this file to test your code. Write the name of the file that contains your function in the appropriate line. **Note that this file tests only for some use-cases. Passing all the test cases in this file may not mean your code is 100% correct.**

HINT 1: You can use **zip** function to **unzip** list of tuples. Use it with unpacking operator *.

```
#customers will contain the first elements in the tuples (customer names)
#bids will contain the second elements in the tuples (bidding prices)
  customers, bids = zip(*bid_list)
```

HINT 2: To sort bid list, you may want to define a function as a sort key. Search online to learn how to do that. You can add more functions to your code if needed.

Input:

1. **bid_list**: A list of tuples that include customer name (string) and the bid (float). You can assume that the length of **bid_list** is at least 2. Example:

```
 bid_list = [('Customer0', 5995.97), ('Customer1', 8800.23),('Customer2', 9912.67),
('Customer3', 4350.68),('Customer4', 6889.54),('Customer5', 10417.86)]
```

Output:

1. As mentioned above, in case of no winner, return (”No Winner” , -1).

2. Your function should find the lowest unique bid and identify the winner. The winner should be returned as a tuple containing name and the bid in this given order. Example:

   ```
   ('Customer3', 4350.68)
   ```

3. Your program should not return a bid that has negative price quote like (”Customer 11” , -115)

Example Outputs:

```
 bid_list1 = [('Customer0', -10), ('Customer1', 32),('Customer2', 15),
('Customer3', 20),('Customer4', 21),('Customer5', 15)]
print(find_winner(bid_list1) # prints ('Customer3', 20)

 bid_list2 = [('Customer0', -10), ('Customer1', 32),('Customer2', 15),
('Customer3', 32),('Customer4', 43),('Customer5', 15), ('Customer6', 43)]
print(find_winner(bid_list2) # prints ('No Winner', -1)
```