CS 3030: Python
Midterm Spring 2019

Choose and solve one question for each difficulty level. That means that you have to solve one easy question, one medium question and one difficult question.

Make sure to comment things to explain what you are doing. Partial credits may be given for incomplete solutions with **clear comments** regarding how to finish the problem in case you did not finish.

Being "name" your name, "surname" your surname and "num" the exercise number (1 to 9): name your exercises as **midterm_name_surname_ex_num.py**. Then put all together in a folder named **midterm_name_surname**. Compress it as **zip** and make sure to upload it to Canvas **before 9:15 a.m**.

Before submitting, make sure:

- The code is clean.
- Your implementation runs for all the possible cases.

(25 points) (Easy difficulty) Choose and solve one:

1. Create an @uppercase decorator that will always return a string in uppercase letters. If the function that decorates does not return a string, invoke an exception or an assertion and state in a comment which one you have chosen and why.

2. Write a allEven() and allOdd() generators. Like integers() but only yields even or odds values.

3. Write a lambda function to construct the following pattern for any n (in this case n = 8):

```
1
22
333
4444
55555
666666
7777777
```

(35 points) (Medium difficulty) Choose and solve one:

4. Extend the @cache decorator developed in the homework to save the cache to a shelve database. Note that:
   - When a program has a shelf open for writing, no other program should have it open for reading or writing. So, you should pass the opened shelf file to cache as an argument like @cache(shelfFile=yourShelfFile). Make sure to close the file at the end of you python file.
   - Make sure your cache key is a string.

5. Open a txt file and count the number of occurrences of each word. Find the word with more occurrences and write the same file in html with that word in bold (Note: This is bold). Note: count "can't", "don't"... as one word. You can use *example.txt* for testing.

6. Write a Python function that checks whether a passed string is palindrome or not. Note:
   - A palindrome is a word, phrase, or sequence that reads the same backward as forward. Some examples you may try:
     - "madam",
     - "redder",
     - "race car",
     - "Eva, Can I Stab Bats In A Cave?".
   - If the argument passed is not a string, invoke an exception or an assertion and state in a comment which one you have chosen and why.

(40 points) (Hard difficulty) Choose and solve one:

7. Write a Python generator comprehension to generate perfect numbers. According to Wikipedia: In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. Example: The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128. Note: You don't need to check for numbers further than 8128 as it can take a long time but make sure your generator is able to generate them.

8. Write a Python program to download IMDB's Top 250 data (movie name, Initial release, director name and rating) using Beautiful Soup. The link for that data is: https://www.imdb.com/chart/top.

9. Develop a Calculator using a class. The class must have a sum, subtract, multiply and divide methods that will return the value of that operation as int or float. Those methods have to be decorated with a @print_total decorator that, for example, will print "Total is: 55" before the function returns so you don't have to write that print in each function. Finally, there has to be a run method that will run the calculator.

   By running:

   ```
   calc = Calculator()
   calc.run()
   ```

   The output of the program should be:

   ```
   Select operation.
   1.Add
   2.Subtract
   3.Multiply
   4.Divide
   5.Exit

   Enter choice(1, 2, 3, 4, 5):1
   Enter first number: 25
   Enter second number: 30
   Total is: 55

   Select operation.
   1.Add
   2.Subtract
   3.Multiply
   4.Divide
   5.Exit

   Enter choice(1, 2, 3, 4, 5):5
   Thank you for playing!
   ```