CSC 401 ASSIGNMENT EIGHT

Due Date: Tuesday, Nov. 7th by 11:58 PM

The purpose of this assignment is to assess your understanding of

- Recursive functions
- Recursive thinking

SUBMISSION

- Include your full name as a comment on the first line of your Python program.
- Include the problem number as a comment before each user defined function.
- Code all problems in one Python file(.py) as YourName_HW8.py
- Upload this file to Submissions folder
- Late assignments are not accepted. To earn partial credit you must submit the work you completed by the deadline.

REQUIREMENTS

Note: You may not use Python statements, functions, data type, etc. that were not discussed in the reading assignment or the lecture notes/videos for this week or previous weeks. This is a class for students who have not programmed before and I expect everyone to code on the same level. If you have a better way of writing the code, then upload two versions: one that codes according to the specifications and the other that demonstrates advanced programming techniques.

All functions must be recursive. You should not have more than 10 lines of code in each function. Each function should have at least one return. Do **not** use any type of loop in any of the functions.

PROBLEMS

1. (10 points) Write a recursive function **square (n)** that takes as input a number and squares the number (raises to the 2nd power) until the result is greater than or equal to 1000, then returns the result.

For example:

```
>>> square(6)
1296
>>> square(-6)
1296
>>> square(6.1)
1384.5840999999996
>>> square(10)
10000
>>> square(1233)
1233
```

2. (10 points) Write a recursive function **alt (s, t)** that takes as input two strings and mixes string s and t with alternating characters from each string and prints the string. Check that s and t have the same length; if not return immediately.
   For example:

```
>>> alt('good', 'bye')
>>> alt('hello', 'world')
hweolrllod
>>> alt('abc', '123')
a1b2c3
```

3. (10 points) Write a recursive function **stars (n)** that takes as input a non-negative integer and generates the pattern of stars shown below for any n.

```
>>> stars(5)
*****
****
***
**
*
*
**
***
****
*****
>>> stars(2)
**
*
*
**
```

4. (10 points) Write a recursive function **prompt ()** that asks the user to enter a password. If the user simply hits return without entering anything, the function should prompt again, until the user enters a string. The string is then returned (not printed).
   For example:

```
>>> prompt()
Enter password:
Enter password: hello
'hello'
>>> prompt()
Enter password:
Enter password:
Enter password: good123bye
'good123bye'
>>> prompt()
Enter password:
Enter password:
Enter password:
Enter password: hi45*70#?gi9
'hi45*70#?gi9'
```

5. (10 points) Write a recursive function **check (s)** that takes a string representing a password as input and prints all the characters that are digits (0 -9). Note the order in which the digits are printed on one line in the following examples.

```
>>> check('hi45*70#?gi9')
9 0 7 5 4
>>> check('abc')
>>> check('34567')
7 6 5 4 3
```

IF YOU HAVE ANY QUESTIONS REGARDING THIS ASSIGNMENT, PLEASE POST THEM TO THE ASSIGNMENT EIGHT DISCUSSION FORUM