CMSC 201 Final Review Sheet 2

1. Circle valid python variable names (some don’t follow coding standards, but are valid)

   1Direction4Ever **ILOVECMSC201** **_num_fru!ts** **DoGsRgOoD**

   thebestclassis201 **print** **Go0D1ucK0NtH#f|nAL!**

2. What gets printed? Does the following code snippet cause an error? Why or why not?

```
my_list = [“here”, “are”, “some”,
           “strings”]
x = 4
if  x < len(my_list) and len(my_list == 5):
     print(my_list[x + 1])
```

3. What gets printed on lines 8 and 10 and why?

```
1     def do_something(my_string):
2         my_string = my_string.upper()
3         my_list = my_string.split()
4         my_string = ””.join(my_list)
5
6      if __name__ == “__main__”
7          my_string = “hello world”
8          print(my_string)
9          do_something(my_string)
10         print(my_string)
11
12
```

4. Describe the difference between for and while loops.

5. Why do we use Boolean flags?

6. Why important to close the file after using it during file I/O?

7. What is the difference between appending and writing to a file?

8. What are the components of a dictionary? Describe their properties

9. What happens when a function is called?

10. Describe mutability, which data types are mutable/immutable, and applications of mutable data types.

14. What is incremental development?

15. Implement a recursive Fibonacci function

16. Implement a pascal’s triangle function.

17. Why would you use a dictionary over a list?

18. What is the output of this code snippet?

```
 def count_vowels(word):
     vowels = [“a”, “e”, “i”, “o”, “u”]
 if word = “”:
     return 0
 elif word[0] in vowels:
     return count_vowels(word[1:]) + 1
 else:
     return count_vowels(word[1:])

if __name__ == “__main__”
   word = “Elephants Are Great”
   print(“The Number of Vowels is ”, count_vowels(word))
```

19. Given the following code, write the output

a. fact = “201 has the Coolest Professors, shhh!”
print(fact[4:6] + fact[21] + fact[33:35])

b. fact = “201 students will do great on the exam if they try hard!”
print(fact[4:8]+fact[46])`

c. fact = “Finally, the Important Things In Life That Matter!”
print(fact[0:5]+fact[28]+fact[43:])

20. What is the minimum number of base cases required for a recursive function? Minimum for recursive cases?

21. What is the correct order for the `range()` parameters?
    A. start, step, stop
    B. start, stop, step
    C. step, start, stop

22. Why can't you iterate through a dictionary with a loop? What can you use to iterate over a dictionary?

23. What is the difference between sentinel values and boolean flags?

24. Describe the best-case runtimes (and why) for the following: bubble sort, binary search, linear search, selection sort, insertion sort, quick sort

25. In some situations, a recursive function will run until a `RecursionDepth` error occurs. *Why* does this error occur, and *what* should be done to fix it?

26. List the fundamental differences between looping and recursion

27. Explain the differences between **`read()`**, **`readline()`**, and **`readlines().`** Give an example of when you might use each.

28. Recursively determine if a number is prime.

29. Recursively determine if a number is a power of n.

30. List and explain the different file access modes.

31. Why do we care about runtimes?

32. What's the difference between top-down and bottom-up development?

33. List string escape sequences.

34. List and differentiate the different ways to access the keys of a dictionary.

35. Rank the following runtimes from **fastest** to **slowest**

    a. n^2, 1, $\log_2 n$, n, $n\log_2 n$

36. Convert the following decimal numbers to binary and hexadecimal:

a. `463`
b. `63`
c. `31`
d. `255`

37. Convert the following binary numbers to hexadecimal:

a. `1010 0011 0101 1111`
b. `1101 1100 1011 0000`
c. `0000 0001 0010 0011`
d. `0110 1011 0101 1011`

38. Convert the following hexadecimal numbers to binary and decimal:

a. `14AD`
b. `002F`
c. `10BA`
d. `FFFF`
e. `13EC`