CMSC 201 Section 40
Spring 2022

Sample Exam #2 ANSWER KEY

Note: this exam is intended to prepare students for the actual second midterm, which will be given on Wednesday, April 20. The type and numbers of questions on this exam are the same as will be on the actual exam.

There will be a separate file containing the answers to this sample exam posted at a later date.

Section 1: True/False and Multiple Choice. This section is worth a total of 30 points. There are 10 questions worth 3 points each. No partial credit will be given for questions in this section.

1. Which of the following statements about reading in a text file in Python is true?
   a. All data come in as strings
   b. You can read the entire file in at once as a single string using the “read” method
   c. Both of the above are true
   d. None of the above is true

2. True or False: for a recursive function to work in Python, there must be exactly one base case and one or more recursive cases.
   a. True
   b. False

3. Which of the following is true about keys and values in a Python dictionary?
   a. **Keys must be unique, while values can be repeated**
   b. Keys and values must both be unique
   c. Keys must be mutable types; values must be immutable types
   d. Both keys and values must be mutable and can be repeated

4. How do you create a new, empty, 2D list, l?
   a. l = [][]
   b. l = {}
   c. l = []
   d. l = [[]]

5. What is the binary number 0b1001 equal to in decimal?
   a. 19
   b. 17
   c. 9

d. None of the above

6. True or false: a dictionary can be a value associated with a key in another dictionary. That is, if d is a dictionary, you can define a new dictionary d-prime such that
d-prime = {“words”: d}

a. True

b. False

7. True or false: a function parameter can be a mutable type, such as a list?

a. True

b. False

8. What is the hex equivalent of the binary number 0b1001100000001111

a. 0x1001

b. 0x980f

c. 0xa80f

d. None of the above is correct

9. True or False: When writing to a file in Python, you can only write strings. You cannot write integers, floats or Booleans without first casting them.

a. True

b. False

10. True or False: every problem that can be solved with a recursive function can also be solved iteratively, but the reverse is not true.

a. True

b. False

Section 2. Short answer. This section is worth a total of 40 points. There are 8 questions, worth 5 points each. Partial credit WILL be given for questions in this section.

11. What is a base case in a recursive program?

A problem so simple it can be solved directly.

12. Suppose you had the following Python program:

```
def print_values (nums):
    for i in range(len(nums)):
        nums[i] += 1
        print(nums[i])

if __name__ == "__main__":
    integers = [1,2,3,4]
    print_values(integers)
```

```
print(integers[0])
```

What is the result of the last print statement - print(integers[0])?

2

The reason is because of how Python handles parameter passing. Since the argument/parameter is a mutable type - a list - the function can directly change the values that the main program sees. This would not be possible with an immutable parameter.

13. Suppose you had a two-dimensional list, l, in which every row was the same length. How would you find the number of rows in l; and how would you find the number of columns in l?

Rows - len(l)

Columns - len(l[0])

14. What happens when you don't specify the mode in a file open statement? E.g., if you wrote

```
with open ("data.txt") as infile:
```

what would happen?

Python defaults to opening for read - "r" mode.

15. What is the difference between opening a file in write or "w" mode and opening a file in append or "a" mode?

"w" mode - all previous contents are erased; start with a new empty file

"a" mode - keep the previous contents; start writing at the end

16. You are given a dictionary, d, containing the teams in the NFL's AFC North division and their starting quarterbacks from last season. That is:

```
d = {'Ravens':"Jackson", "Steelers":"Roethlisberger",
"Bengals":"Burrows", "Browns":"Mayfield"}
```

What Python statement(s) would you use to determine if Tom Brady is one of the quarterbacks in the AFC North?

if "Brady" in d.values():

17. Suppose you wanted to write Python code that would put the first 25 integers - from 0 through 24 - into a two-dimensional list with 5 rows and five columns. Write code that would do that for you. You do NOT have to write an entire program; just the lines of code necessary to put the numbers in the 2D list.

```
num_list = [ ]
for i in range(5):
    new_row = []
    for j in range(5):
        new_row.append(i*5 + j)
     num_list.append(new_row)
```

18. Suppose you have the dictionary

```
    senators =  {"Maryland":"Cardin",
"Virginia":"Warner","California":"Feinstein", "Florida":"Rubio"}
```

Describe what happens when you execute the statement

```
  print(senators.get("DC","not a state")
```

"not a state" is printed.

For dictionaries, "get" returns the value associated with the specified key. If that key is NOT in the dictionary, "get" returns whatever you put in the "get" statement, or None if you don't specify anything. In this case, "DC" is not a key in the dictionary, so "get" returns the string "not a state" and that's what gets printed.

Section 3. Programming. This section is worth a total of 30 points. There are two problems worth 15 points each. Partial credit WILL be given for problems in this section.

19. Write a recursive program that calculates the sum of the digits in a positive integer. That is, given an input of 12345, the program would print the value 15; while given an input of 123 the program would print 6.

The program should contain one recursive function, and a short main program that gets the integer to sum and then calls the function.

Hint: think about what your base case is. Then think about how you would remove the last digit from the number for a recursive call.

```
def sum_integers(num):
    #base case: the sum of a one-digit
integer is that integer
```

```
    if num == num%10:
        return num
    #recursive case: integer divide by 10,
which removes the last digit
    # add that digit to whatever's returned
by your recursive call
    else:
        d = num % 10
        rest = sum_integers(num//10)
        return d + rest
    # you could also just write this as:
return num%10 + sum_integers(num//10)

if __name__ == "__main__":
        x = 123
        print(sum_integers(x))
```

20. Suppose I have a file, cities.txt, that contains the names of the five largest cities in Maryland with their population. The file looks like this:

Baltimore	736,014
Frederick	40,148
Rockville	44,830
Gaithersburg	39,676
Bowie	37,642

That is, there is one city per line. Each line has the name of the city, then a tab, then the current population of the city. Write a Python program that reads in this file, computes the arithmetic mean - the average - population of these cities, and prints that number out.

- You can assume that this file is in your current directory; you don't need to provide a path
- You do not need to write functions; you can write the entire program in the main program if you want.

```
#read and prep data
with open("cities.txt","r") as infile:
    data = infile.read();
    city_list = data.split("\n")
    for i in range(len(city_list)):
        city_list[i] =
city_list[i].split("\t")
        city_list[i][1] =
int(city_list[i][1])
    #now compute average
    sum = 0
    for i in range(len(city_list)):
        sum += city_list[i][1]
    average = sum/len(city_list)
    print("The average population is ",
average)
```