CMSC 201 Section 60

Fall 2021

Sample Exam #2

Answers are in red

Note: this exam is worth 125 points, weighted as follows:

Multiple Choice & True/False: 40 points – 10 questions, 4 points each, no partial credit

Short answer: 40 points – 5 questions, 8 points each – partial credit is given

Programming: 45 points – 3 questions; 15 points each – partial credit is given

Section 1: T/F – M/C Section.

1. True or False: A recursive function in Python can have two separate recursive cases.
True - it just has to have at least one. More than one is fine.

2. True or false: the best way to develop a complex computer program is to write the code for several functions at once, then test them at the end. There's no need to test as you go if you never make mistakes, is there? False

3. The difference between opening a file for writing ('w' mode) and opening a file for appending ('a' mode) in Python is: a. Is the correct answer

a. Opening for writing erases all previous contents of the file, while opening for appending keeps the previous contents and starts writing at the end

b. If the file does not exist, 'w' mode creates a new file for you while 'a' mode returns an error message saying the file doesn't exist

c. 'a' mode allows you to write strings, integers, floats and Booleans to the file while 'w' mode only allows you to write strings

d. 'w' mode only allows you to write to the file, while 'a' mode allows you to both read from and write to the file.

4. Suppose I had the following dictionary defined in a Python program:

senators = {"New York":"Schumer", "Maryland":"Cardin", "Kentucky":"Paul", "Virginia":"Warner"}

How could you insert Kristen Gillibrand as another Senator from New York WITHOUT deleting the entry for Chuck Schumer? B is the correct answer

a. senators[“New York”] = “Gillibrand”

b. senators[“New York”] = [“Schumer”, “Gillibrand”]

c. senators.insert(“New York”:”Gillibrand”]

d. senators.get(“New York”, “Gillibrand”)

5. If you have a Python 2D list, l, that has r rows. Which of the following is a true statement? D. is the correct answer. A is wrong because len(l) yields the number of rows in the list. B is wrong because you would use l[i][j]. C is wrong because it’s exactly backward - you can manipulate a row at a time with a single statement, but columns can only be manipulated in loops.

a. len(l) yields the number of columns in each row

b. if you want to access the element in row i, column j, you would use l[I,j]

c. You can manipulate an entire column at a time with a single statement, but rows can only be manipulated in loops

d. None of the other statements is true.

6. True or False – if there is a return statement in a function, but it never gets executed, the function has no return value. False. The return value would be None. All functions always have a return value.

7. Suppose we have the following program:

```
Def find_area(base, height): #these are the parameters

        Area = (base * height)/3

        Return area

If __name__ == “__main__”:

        Height = 3
```

Radius = 4

Volume = find_area(height, radius) #these are the arguments

Print (volume)

What value gets assigned to the parameter "height" when the function find_area is called? B is the correct answer. The parameter means the occurrence of "height" that's in the function, not the one in the main program

a. 3

b. 4

c. 2 * pi / 3

d. None of the above

8 What happens if you do not specify a mode in your file open statement in Python? That is, suppose your command is:

with open("data.txt") as myfile: b is the correct answer

a. This will be an error and your program will crash

b. Python defaults to opening the file for reading

c. Python defaults to opening the file for both reading and writing

d. None of the above answers is correct

9. True or false: every problem that can be solved with recursion can also be solved iteratively (without using recursion), but the opposite is not true. True

10. True or false: When you read in a file using read(), readline(), or readlines(), everything you read in is treated as a string or a list of strings. True

Section 2: Short Answer Section

11. Suppose we have the following dictionary defined:

```
governors = {"Maryland":"Hogan", "Virginia":"Northam", "Louisiana":"Edwards",
"Massachussetts":"Baker"}
```

How would I check to see if the Governor of California was defined in the dictionary, if I wanted to make sure that the program would NOT crash if Gavin Newsom wasn't there? You can either answer with a line or two of code, or with a prose answer.

```
if governors.get(“California”, -1) != -1:

        print(“California is there”)

else:

        print(“It’s not there”)
```

12. Write a Python function that calculates the volume of a right circular cone, which is given as v = the height of the cone times pi times the square of the radius, divided by three. Presume the function takes two parameters, height and radius, which are floats. The function returns the volume, which is a float. You do NOT need to write a main program; just the function will do.

```
def volume (height, radius):

        return(height * (radius**2) * 3.14159)

         #any number of decimal places for pi would be acceptable; you don’t need 5
```

13. Here's a recursive function that calculates the sum of the digits in a number, which is passed in as a parameter:

```
Def sum_digits(number):

        If number < 10:

        Return number

        Else:

        This_number = number %10

        New_number = number // 10

        Return this_number + sum_digits(new_number)
```

Using the concept of the stack frame, explain how Python knows exactly what value of this_number to use at any given time.

Every time a function is called, Python pushes a new frame on the execution stack that contains the variables, constants and other symbols known right now. When a statement containing “this_number” is executed, Python will look in the frame at the top of the stack and use THAT location in memory, and whatever value is there.

14. Write code that uses the split() and join() functions to change the string “Probably Secure Operating System” to “Provably Secure Operating System”.

```
s = “Probably Secure Operating System”

list_s = s.split()

list_s[0] = “Provably”

s = “ “.join(list_s)
```

15. Suppose you wanted to create a 2D list of the first 100 integers – from 0 through 99 - in 10 rows and 10 columns. Write code that will create such a 2D list.

```
integer_list = []

for i in range(10):

    temp_list = []

    for j in range(10):

        temp_list.append(10*i + j)

    integer_list.append(temp_list)
```

Section 3: Programming Section.

16. I have a file, cities.txt, containing a list of the ten largest cities in the world and the population of each. There is one entry per line, and the name of the city is separated by a tab from the population. The file looks like this:

```
Tokyo	37400000

Delhi	28514000

Shanghai		25582000

Sao Paola		21650000
```

```
Mexico City            21581000

Cairo   10076000

Mumbai                 19980000

Beijing  19618000

Dhaka   19578000

Osaka 19281000
```

Write a Python program that reads in this cities.txf file and calculates the arithmetic mean – the average – of the city populations. Print out that number, with an appropriate label. You can presume this file is in your current directory or path; you do not need to specify a path to it. You must write the entire program. You do not need to use functions; you can do everything in the main program if you wish.

```
with open(“cities.txt”, “r”) as infile:

    data = infile.read()  #read the file as a single string

    city_list = data.split(“\n”) #split into a list, throwing away new-line characters

    total_pop = 0

    for i in range(len(city_list)): #split each row into two columns

        city_list[i] = city_list[i].split(“\t”)  #because I told you there were tabs

        total_pop += int(city_list[i][1])  # add the population to the total

    ave_pop = total_pop/len(city_list)

    print(“The average population of a city is “, ave_pop)
```

17. Write a Python program with a recursive function that determines the number of characters in a string. Yes, we know you can determine the number of characters in a string by calling the len() function, but that’s not the point. 😊 Write the full program – main program and recursive function. Be sure to include both the base and recursive cases.

```
def num_chars(s):

    if len_s == 0:
        return 0

    else:

        return 1 + num_chars(s[1:])

    # note the above chops off the first character in the string to make the call. If you

    # want to chop off the last character it would be return 1 + num_chars(s[:-1])

if __name__ == "__main__":

    s = "whatever string you want"

    r = num_chars(s)

    print("string ", s, " is ", r, " characters long")
```

18. Write a Python program that fills a dictionary by taking input from the user. Your program should initialize an empty dictionary. Then enter a loop where you ask the user for a key, then a value; and then you enter the key:value pair into your dictionary. Stay in this loop until the user types "Q" as the key

```
d = {}   #initialize an empty dictionary
next_key = input("Please enter the next key; enter Q to quit")
while next_key != "Q":
        next_value = input("Please enter the value for that key")
        d[next_key] = next_value
        next_key = input("Please enter the next key; enter Q to quit")
```