CS 1030 – Spring, 2020 – Exam #3 – Total 100 Points

The test is worth 150 points on Blackboard; score below is multiplied by 1.5

Name Joshua Hoshiko Section 3

***Read instructions and specifications carefully!!*** You have 2 hours for the exam. This is a 5-page, no electronics exam except to use your computer for the exam. No Internet access except for our communication. You can use the Python Programming Study Guide (famous file #62). Write all answers in this exam file. Name the file FirstnameLastname\_3 \_Test3. Failure to name the file correctly results in a loss of 5 points. The first part of the exam is Python programming worth 50 points. The second part contains two essay questions worth 50 points.

**PART 1: WRITE PYTHON PROGRAMS**

Problems are adapted from *The Python Workbook*, by Ben Stephenson.

1. (10 points)

Create a program that reads the name of a month from the user as a string. Your program should display the number of days in that month. For February, display “28 or 29 days” to allow for leap years. You must use lists or tuples (and not dictionaries) to hold the names of the months and the number of days in months.

month\_names = ["january", "february", "march",

"april", "may", "june",

"july", "august", "september",

"october", "november", "december"]

month\_days = ["31", "28 or 29", "31", "30", "31", "30", "31", "31", "30", "31", "30", "31"]

from\_user = input("Please enter the name of a month: ")

month\_index = 0

for month in month\_names:

if from\_user.lower() == month:

print("There are", month\_days[month\_index], "days in", month)

month\_index += 1

2. (15 points) Read presses just the <Enter from the user a list of words, one word at a time until the user > key. Your program will output a word or words with commas inserted at the common positions in a list of words. For example:

Apples

Apples and oranges

Apples, oranges and bananas

Apples, oranges, bananas and kumquats

Your program will be able to process any number of words starting with one word. (The user won’t press just the <Enter> key when the program starts.)

word\_list = []

while True:

from\_user = input("Please enter a word. Just press the <Enter> key to continue: ")

if from\_user == "":

break

else:

word\_list.append(from\_user)

if len(word\_list) == 1:

print(word\_list[0])

else:

print(word\_list[0], end="")

counter = 1

while counter != (len(word\_list) - 1):

print(", " + word\_list[counter], end="")

counter += 1

print(" and", word\_list[counter])

3. (25 points) Using a dictionary, create a program that determines and displays the number of unique characters in a string entered by a user, and the number of times each unique character appears. For example, the string Hello, World! yields a count of 10 unique characters and a dictionary output of { ‘H’ : 1, ‘e’ : 1, ‘l’ : 3, etc. }. The order of the output does not matter. A string of zzzz yields a count of 1 and a dictionary output of { ‘z’ : 4 }.

output\_dict = {}

from\_user = input("Please enter a word or phrase: ")

char\_counter = 0

for char in from\_user:

if char not in output\_dict:

output\_dict[char] = 1

char\_counter += 1

else:

output\_dict[char] += 1

print("Number of unique characters:", char\_counter)

print("Output dictionary:", output\_dict)

**PART 2: ESSAY QUESTIONS**

4. (25 points) Here’s a slight variation of what’s known in ethics circles as “The Trolley Problem.” Here’s the scenario: a driverless car without a human driver has a brake failure and is headed towards a group of pedestrians. Ahead there are four 80-year-old women to the left and a 25-year-old woman pushing a baby carriage with a baby visible to the right. There’s a 4-foot high cement divider ahead in the middle of the road. What should the driverless car do? Answer in a 3-5 sentence paragraph and justify your answer. Then, in a 3-5 sentence paragraph, take the opposite point of view and justify that answer. If you write more than what is requested, I will deduct five points. The purpose of this limitation is for you to show that you can summarize well. (By the way, in ethics circles, a trolley takes the place of the driverless car.)

There are many metrics that the driverless car can measure to decide the best decision, but most of these metrics are arbitrary. We could decide based on the occupation of the individuals and their estimated contributions to society, we could decide based on age, we could decide via quantity of individuals, or perhaps we decide based on an estimated chance of survival for individuals. None of these metrics are objectively right, but let’s say we measure life by contribution to society. It is feasible that the elderly may be estimated to have less contributions because they are already aged and perhaps retired. The Algorithm in the car might decide to head that way because a child and young mother still have a lot of potential to contribute to society. This would be a judgement based on age and contribution metrics.

However, let’s say the car decided based on quantity or chance for survival/least collateral damage. The car then might swerve towards the mother and the child. This is because the mother is younger and can probably survive surgery and perhaps the carriage could potential protect the child. There are also less individuals in this path. Alternatively, does the car just prioritize pedestrians and ram into the cement killing the passengers? Well, then we also have to analyze the passengers. There are many angles one could take but these are just a few.

5. (25 points) During the semester we experienced a wide range of subject matter through videos, class discussions and reflection assignments. Pick a *controversial* topic from any of them, take a position (there could be many) and write 3-5 sentence paragraph justifying your position. Then, take an opposite position and write a 3-5 sentence paragraph justifying the opposing position. If you write more than what is requested, I will deduct five points. The purpose of this limitation is for you to show that you can summarize well.

When it comes to the judicial system, a topic I found very interesting was the use of algorithms to determine sentence times. One important factor to consider is how unpredictable humans can be when judging others, where simple mood swings can have effect on the judge’s decisions. However, algorithms are consistent and can find patterns that we do not see, meaning that an algorithm could potential make a more informed decision based on metrics we do not initially consider. With these algorithms improving, we could have breakthroughs in the form of improved psychological analysis and more accurate sentencing to heal/correct an individual.

However, it is easy to put too much trust into an algorithm and this can result in errors that nobody catches because they blindly trust the algorithm. There are many examples of absurdly harsh sentences that made it through court because the humans blindly trusted the algorithm. The best approach is to layer several systems together that all check one another. Perhaps multiple bots layered together to offer different perspectives or human layers integrated with machine layers. Better metrics and analysis of the algorithm’s decisions can help prevent errors and all the layers working together can help prevent the errors of both machine and human.