Due: 02 May 2024, 23:00, to Submit

ASSIGNMENT3

BBM103

National Intelligence Agency

Instructors: Dr. Tunca DOĞAN
TA: Hayriye ÇELİKBİLEK
Course: BBM103 Spring 2024
Subject: File I/O and Dictionaries
Given: 25.04.2024
Due: 02.05.2024, 23:00, to Submit

Introduction

The Submit system is our department's assignment management web application for students and instructors.

Since we handle student and instructor needed requirements privately using our services, rather than having the requirements only for developing some software, we will collect the following assignments (excluding this assignment) using the Submit webpage.

Policy

- Be sure to complete the submission deadline. The deadline is 23:00, and you see 23:59 on the live because of the advance compensation of potential problems. Last-minute excuses will not be tolerated.
- Save all your work until the assignment is graded.
- You can ask your questions via Piazza, and you are supposed to be aware of everything discussed on Piazza.
- You must submit your work using the file hierarchy as stated below.
- No other submission methods will be accepted (mail)

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss the given assignments with your classmates, but these discussions should be carried out abstractly. Discussions about a particular solution to a problem (either in actual code or pseudocode) will not be tolerated. In short, you are turning in someone else's work (from the internet), in whole or part, as your own will be considered a violation of academic integrity. The former condition also holds for the material on the web, as everything on the web has been written by someone else.

References for the Academic Integrity (AI):

https://academicintegrity.ucsd.edu/AI-Handbook-for-UCSD_2019.pdf,

academicintegrity.org/resources/facts-and-statistics

Joining Brilliant

Brilliant is an online learning platform that focuses on STEM-related topics. The 60+ courses on Science, Technology, Engineering and Math topics all offer an **interactive and hands-on learning experience**.

It is mandatory to use your **CS e-mail address** as your account, your student ID b2XXXXXX as your name and the text '.' (dot) as your surname while signing up (see Figure 1).

It is a must to use the invitation link to let course instructors see your solutions and time on the platform.

https://brilliant.org/classroom/join/yzaxce/

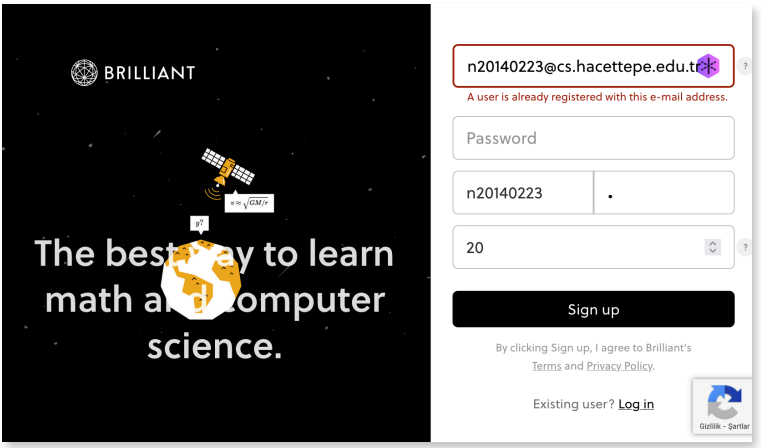

Figure 1. How to join brilliant.org

Work To Be Done

You are responsible for solving 8 lessons from Creative Coding in the Brilliant Classroom (see Figure 2).

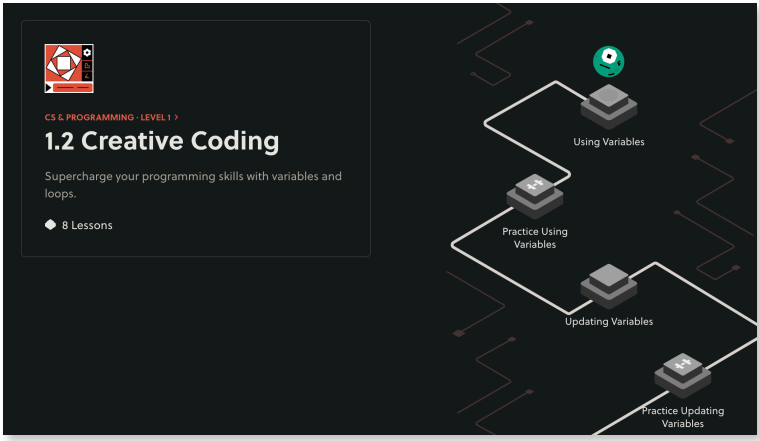

Figure 2. The overall display of the lessons.

If you cannot solve some of these examples correctly, do not worry; the interactive problems include the solutions if you make any mistakes. There will be no point deduction about the spent time and correctness in Assignment 1; we value your effort. Take your effort wisely.

These lessons should take 5 to 15 minutes each, and the problems are fundamental and similar to your BBM103 slide examples, so you do not have to worry.

Security Counsel

You are a brilliant and competent National Intelligence Agency agent. You are working on a department that researches scientific interrogation techniques to discover the real bombers (See Figure 3). Your agency has a confidential interrogation technique / algorithm to counsel if the interrogee is a bomber. However, the current algorithm has 99.99% confidence level for its final counsel. Also, previous versions of the algorithm had less than 99.99% recognition accuracy.

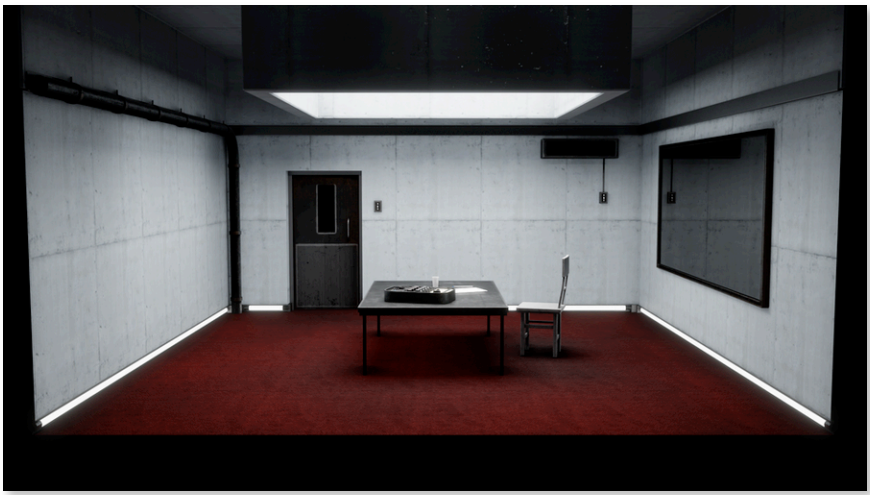

Figure 3. Footage of the Interrogation Room.

COMMANDS

For this prediction utility Interrogee Name, Core Accuracy, Counsel, Local Bomber Incidence, Counsel Risk

will be given an input file named "agents_aid_input.txt" an example of this file is provided in the attachement. Note that the agent does not have to give the commands in the same order here. Each type of command can be used in any order in the input.

Command types in the input file are as follows:

- **Create a new Interrogee:** The "create" command will add a new interrogee to the system (in RAM). For example, "**create Hayriye, 0.999, 0, 50/100000**" command means that an interrogee named Hayriye has been diagnosed with a "**not bomber**" counsel using the confidential interrogation algorithm that has a **99.9% accuracy ratio**. The local population in the city has 50 bombers for every 100.000 citizen; this phenomenon is known as incidence. **The algorithm suggested a release**. The output of this command will be "Interrogee Hayriye is recorded." For another case, the output will be "Interrogee Hayriye cannot be recorded due to duplication." You do not have to check for other possibilities. **Please note that the input data are artificially produced. They are not indicating real-world scenarios.**

- **Delete an existing Interrogee:** The "remove" command will remove an existing interrogee from the system. For example, "remove Deniz" means the agent wants to delete the interrogee named Deniz. The output of this command will either be "Interrogee Deniz is removed" or (if there is no such person) "Interrogee Deniz cannot be removed due to absence.".

✤ **List all Interrogees with their Information:** The "list" command will be used to list the complete information in the system. For example: "list". Note that there is no additional operand for this command.

Interrogee Name	Core Accuracy	Counsel	Local Bomber Incidence	Counsel Risk
Hayriye	99.99%	Not Bomber	50/100,000	No Risk
Pakize	95%	Not Bomber	45/100,000	No Risk
Muzo	99.99%	Bomber	50/100,000	0.1666
Deniz	99.9%	Bomber	40/100,000	0.6668
Toprak	95%	Bomber	21/100,000	0.9960

✤ **Counsel Risk:** The "risk" command will be used to calculate and show the actual probability of a counsel to be false. This is fatally important to decide if the person is a bomber or not. For example, "risk Deniz" command will have an output of "Interrogee Deniz has a counsel risk of **66.68%**." (**Yes it is true! Even though the algorithm accuracy is 99.9%**) (See also: FP)(∓0.02% is OK in your prints). This weirdness is called **base rate bias** or **conditional probability**. This calculation is made according to the core accuracy and incidence information given above in the creation part. If the risk is less than 3%, the output will be "Interrogee Hayriye has no counsel risk." For absence "Risk for Ateş cannot be calculated due to absence."

✤ **Recommendation for a particular Interrogee:** The recommendation command will be used to give a system recommendation to the agent about whether to

release the person. As you might notice the decision will be based on Counsel and Counsel Risk features. And it is really tricky! Even if it seems everything signs bomber for Deniz and Toprak their risk to be innocent is too high that we need other validation methods than our algorithm to charge them. For example, "recommendation Deniz" command will result in either "System suggests to arrest Deniz." or "System suggests to release Deniz." The recommendation calculation will check counsel column and compare the risk with 40%. If the risk is too high (bigger than 40%) the recommendation will be the opposite of the counsel (algorithm result). This assignment will have no confusion, no equal probabilities, or no missipelled inputs while testing and grading. The other case of the recommendation output could be “Recommendation for Deniz cannot be calculated due to absence.”, if there is no such Interrogee in the system. The only case you should check is when an interrogee name is given (in any command) whether the person exists in the system. You do not have to check for other input correctness conditions.

- Also, while the commands will be provided via a file, outputs of each command’s step will be provided to an output file named “agents_aid_outputs.txt”
- An example use-case test scenario is provided as "agents_aid_outputs.txt" and "agents_aid_inputs.txt"
- We covered Base Rate Bias (Conditional Probability) in class with a cancer example. You wil use the exactly same calculation here. Cheat Sheat is in the Appendix.

IMPLEMENTATION

Using a dictionary for all interrogee data in the system is obligatory. For example, Hayriye will have the dictionary { "Interrogee Name": "Hayriye", "Core Accuracy": 0.9999, "Counsel": "Not Bomber", "Local Bomber Incidence": "50/100000", "Counsel Risk": "No Risk" } represents the complete information for the particular person. The data have to be saved onto a dictionary in this assignment. The recommendation, on the other hand, will not be calculated on the go, it will be saved like this dictionary. The dictionaries can be stacked into a list or list-like object. Necessary printing operations can be held besides this dictionaries. Deleted interrogees will be removed from this list. Names of the people will be unique, and they can be used as an identifier for the search operations. Also, you should develop a habit of writing explanatory comments during your development for clarity.

You will need to implement at least seven functions:
(i) a recommendation function,
(ii) a risk function,
(iii) a listing function,
(iv) a creating new interogee function,
(v) a removing person function,
(vi) a reading the input file function and
(vii) a saving to the output file function within this assignment implementation.

✤ You must prepare your algorithm flowchart for your functions. A generalized flowchart version to reveal the function and pseudocode design is necessary.

- ✤ Do not use unnecessary global variables (this is why we have functions, to cooperate the design)
- ✤ If your program fails to read some of the inputs, your whole assignment could crash at the beginning! This indicates no grading.

CS Submit System

Finally, submit your **b2XXXXXX.zip** folder with your code, and flowchart inside (see Figure 4) to the Submit web page once completed (see Figure 5). These two files and their naming are obligatory to have.

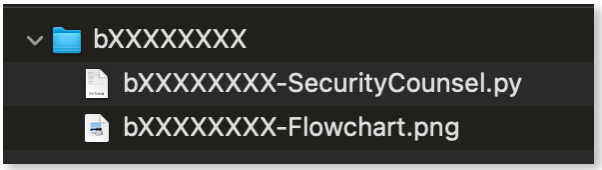

Figure 4. Showing your overall project hierarchy.

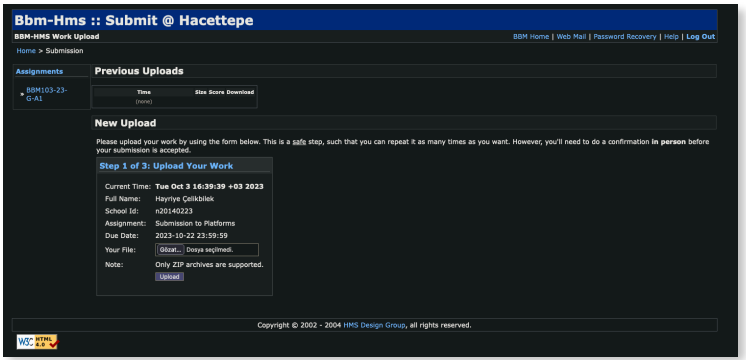

Figure 5. The overall vision of the Submit system.

Grading

In evaluating the assignment, the scoring is as follows:

Evaluation	Points	Evaluate Yourself
Solving 8 lessons from Brilliant	15	
Wrong Folder, File names and hierarchy	-100	
Solving Brilliant without enrolling with the classroom link	-100	
No student ID signup naming in Brilliant	-100	
No student ID in code.	-100	
No student ID in flowchart.	-100	
Code not working	-100	
Missing Submit submission	-100	
Using global variables	-20	
Missing flowchart	-20	
There are many negative rubrics.	-?	
Insufficient comments	-10	
Inefficient solution design	-10	
Flowchart	10	
Recomendation function usage	5	
Risk function usage	5	
Listing function usage	5	
Create function usage	5	
Remove function usage	5	

Evaluation	Points	Evaluate Yourself
Read file function usage	5	
Write file function usage	5	
Test scenarios correctnesses	40	
Total	100	

You may include your evaluation guess as a file in your .zip folder named "self-evaluation-table.pdf" (or any other extension). This self-evaluation table is optional for feedback about your expectations, self-awareness, and effort (see Figure 5).

Figure 6. Interrogation Scene while evaluation! 😳

Appendix

Algorithmic Thinking, Visualization and Flowcharts:

- Python Code Debugger: visualizing code execution: pythontutor.com
- Creating flowcharts from text (pseudocode): chartmage.com/index.html
- Automatically create flowcharts from Python code: https://github.com/cdfmlr/pyflowchart http://flowchart.js.org/
- Creating flowcharts from code-like text: app.code2flow.com
- Creating flowcharts manually: draw.io or app.diagrams.net

Visualizing your mind and code while developing to debug or for reporting purposes is essential!

CHEAT SHEET