TASK 1

In order for someone to run the program what he needs is to have python version 3.x installed on his machine, open the terminal and navigate to the task1.py directory in which there is also the appointments.txt file, then all he has to do is enter the command “python task1.py”.

Since the features and purpose is pretty simple and linear It was decided not to use functions and classes. “Pepper” during the conversation, unless she receives an appropriate answer, she needs to repeat each question in order for her to check-in the visitor to his appointment. She is stuck at the greeting part of the conversation until she gets a proper “hello” from the visitor. After she passes that part the visitor will have to answer her question, if he has an appointment with a “yes” or “no”. It was easily accomplished with some “if” conditions and “while” loops. Now if the visitor answers with “no” she will have to terminate the program after she kindly goodbyes him cause at this stage she is not programmed with more options like make an appointment on her own. If the answer was yes, she will ask him his name. She will only accept two entries (first and last name) and repeat the question otherwise. In addition, she will check the given name if it exists on the database with the appointments by reading the file "appointments.txt" line by line comparing it with two “strings” that were created (one with the “last name” first and one with the “first name” first) in which a substring “visitorName“ was added at the beginning in order to be exactly as the string saved in the database in case it exists. If the name doesn’t exist, she will inform the visitor to give a different name.

While she was checking the name is kept the position it was at the database knowing that the host of this appointment should be one position after (i+1) so after retrieving the name she will ask the visitor if he is the one he has an appointment with. If he is not the one she will inform the visitor that an error must been made and that the visitor needs to make a new one by contacting the personnel or start again the conversation in any case, he was the one who made a mistake. If the name given was right the Pepper informs the visitor that the check in was made by creating (if not existed) the “checkings.txt” file in which she will first check if the check-in was already made or add a new entry to it by adding the person id which is at the spot “i-2” and the current date and moment and that’s how the conversation ends.

Every entry to the terminal is case-Insensitive by adding the “.lower()” function to the strings converting them all to lowercase. Only the “datetime” library was imported to capture the moment the check-in happens. The “appointments.txt” file is open with only the read option available since she doesn’t need to make any changes meanwhile the “checkings.txt” with both read and append option cause she needs both to read it and add a new entry.

If I had more time I would add the feature to create appointments or update and delete ones through this conversation. She could inform the host with a message that the visitor arrived and perhaps sent a reminder if she had the time of the appointment to both host and visitor that the appointment is in a half an hour. She should have a way of herself to contact one of the personnel in case an error occurred or if there are way too many visitors at the lobby.