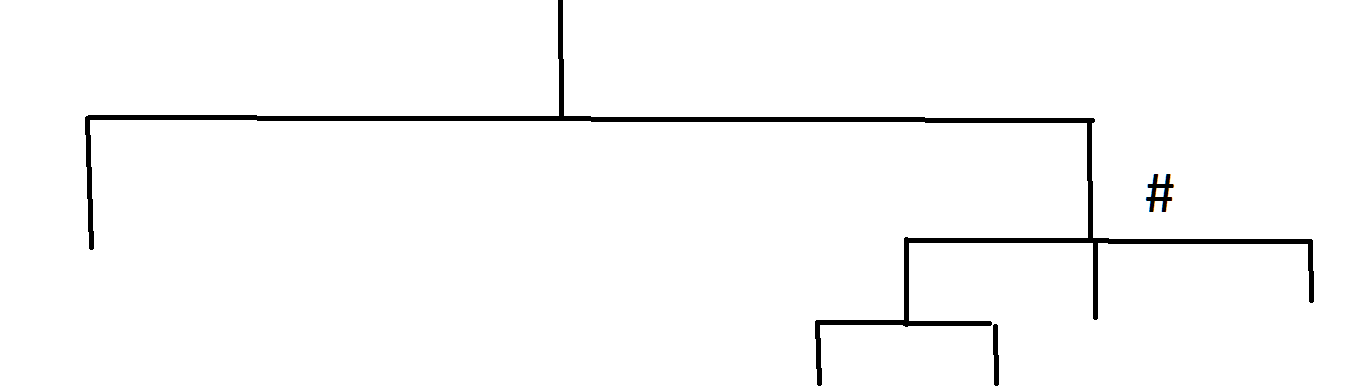
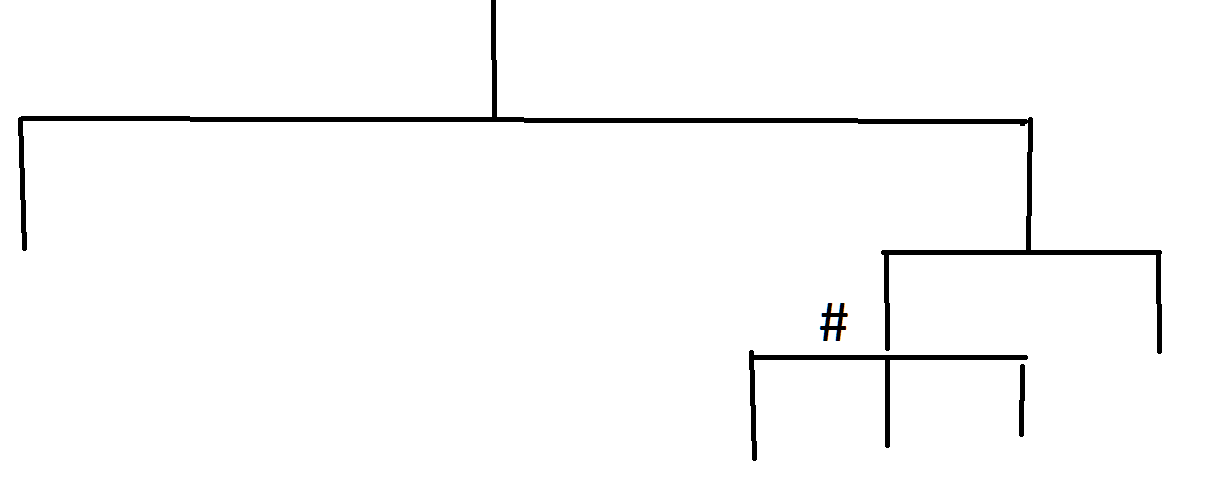
**2. Planning**

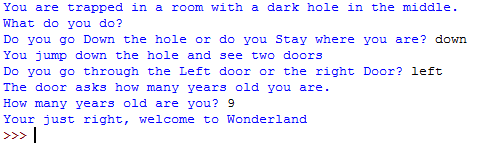
While planning for our code we drew two possible ways that we wanted the gamebook to follow. The drawings looked like this:

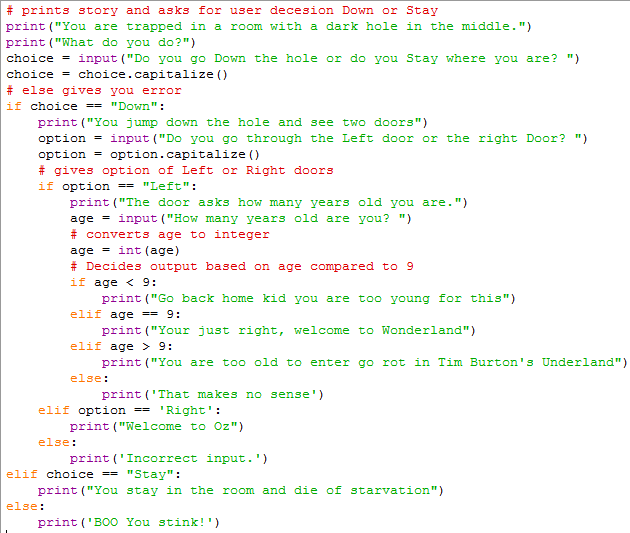


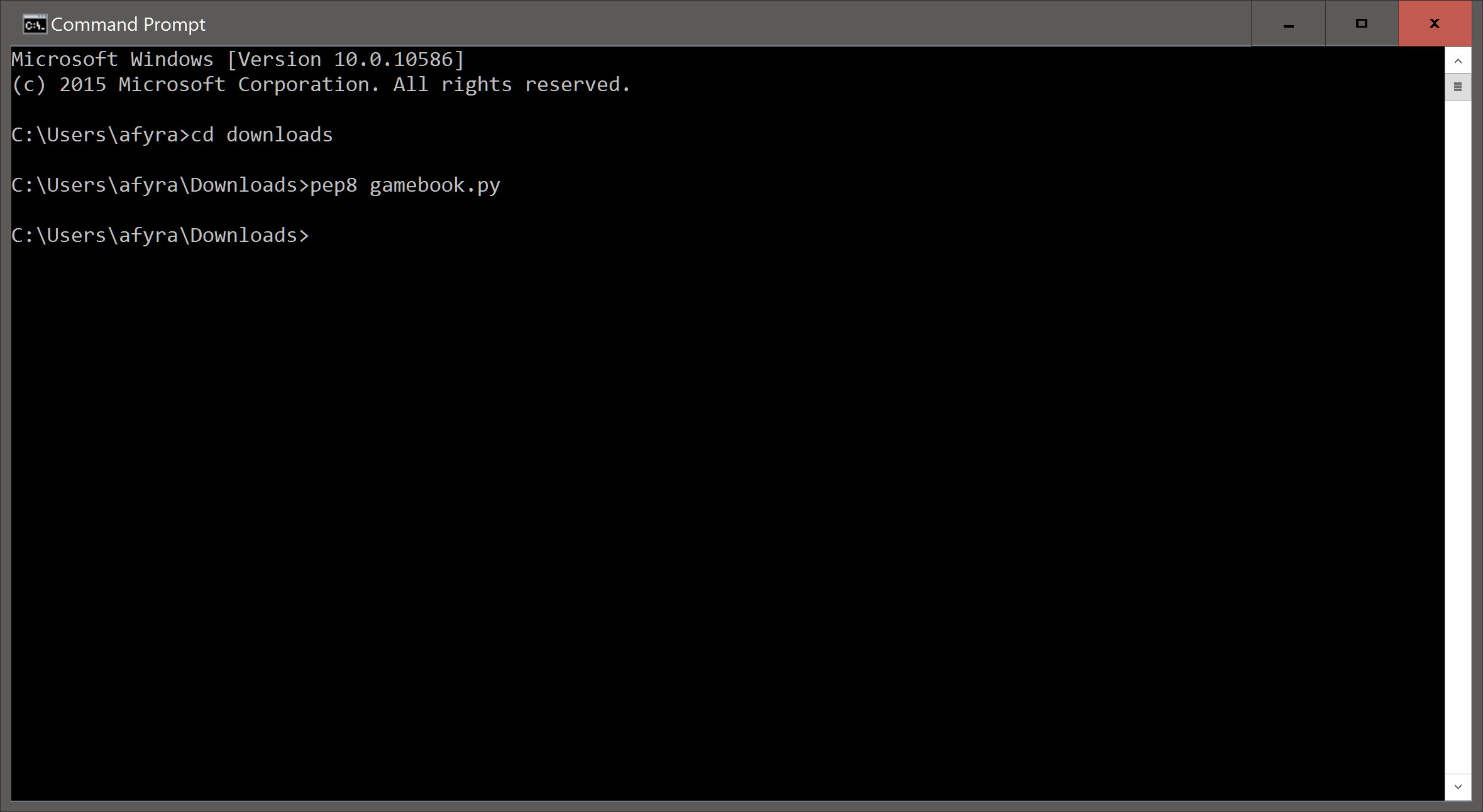
These trees were two possible plans for the code. The # sign on the trees indicate where a numeric decision would be implemented. Upon discussion we came to the conclusion to go with the first tree, allowing us to end on a number based decision.

After picking what path to go down, we needed to come up with a story to follow. Through a short discussion it was decided to go with a story that starts the user in a room with a hole in the middle. This story seemed easy for us to create and have multiple outcomes that can fill the requirments of the lab.

Planning for what code to use was quite simple. We decided to use input commands, print commands, if statements, else statements, and elif statements to satisfy the requirements. This would keep the code simple and easy to fix if errors occur.

**3. Implementation and Testing**





When the code was first ran, a false input would move the user to the else statement. To fix this issue, we changed the else statements to elifs and made else statements to say that the input was incorrect. This will give the user feedback if they enter in an incorrect answer.

Another issue that occurred was when we entered ‘down’ for the first decision instead of ‘Down’. This would cause the user to activate the else statement instead of being put through the if statements. To solve this, after each input we put the command: .capitalize(). This would capitalize the first letter of the decision and allow the user to follow through the if statements.

Running the code through pep8, we found that some lines were considered too long by pep8. To prevent this from occurring we changed option=input(‘blahblahblah’).capitialize() to option=input(‘blahblahblah’)

option=option.capitalize()

This allowed the code to be shortened and in turn made this section of the code pep8 compliant. There still remained a few pep8 errors after this. It just involved the formatting of the comments in the code. We needed to change #blahblahblah to # blahblahblah. After changing this pep8 found no more errors in the code.