To-do list (3 Sep) - Done

* Import starting libraries (bcrypt, sqlite3, tkinter)
* Create starting functions (hash password, main menu, etc)
* Create starting menus (main menu, ‘make maze’ menu, ‘generate maze’ menu, ‘my mazes’ menu)
* Implement Prim’s algorithm for non-perfect maze generation
* Display the maze to the user using tkinter canvas

To-do list (27 Sep) - Done

* Create a data structure (2D Array) to represent the maze.
* Initialise Maze: Initially, all cells can be walls. The algorithm will then carve paths through them.
* Have the program use the prim’s algorithm to carve out the maze
* Rendering the maze by using the data to draw the maze on the canvas.

To-do list (29 Oct) - Done

* When I select “Non-perfect” it will use Prim’s algorithm
* But when I select “Perfect” it uses an algorithm for a perfect maze.
* I don’t have a second algorithm… so I need to first define a new algorithm for perfect maze generation.
* This will be the “Recursive Backtracker” algorithm

To-do list (5 Nov) - Done

* Additionally I would like to add a “Randomize” button to the make\_maze\_menu to randomise the x and y coordinates and instantly click on “Generate maze”

To-do list (6 Nov) - Done

* Next thing, I want to add an ability to “Save” mazes.
* Once a maze has been generated, there should be a button below the maze called “Save”.
* The save button will save the exact maze, and store it in a database.
* When pressing on the “Save” button, there should appear a message “Successfully saved the maze”
* After this, if the user goes back to the main menu, and clicks on “My mazes” there will be a scrollable list that displays saved mazes whilst the program is running.
* Every saved maze appears as a box with the parameters about the maze (e.g. the x and y coordinates), and how long ago it was saved.
* In the box, there will be a button “Generate” that generates the same exact maze that has been saved.

To-do list (9 Nov) - Done

* Implement a method to delete a specific maze from the database.
* Implement a method to delete all mazes from the database.
* Add a "Delete" button next to each "Generate" button.
* Add a "Delete all" button at the top of the "My Mazes" menu with a confirmation dialog.

To-do list (11 Nov) - Done

* I would like to implement a sign up and login system.
* Currently, when I run my program it just straight up opens the program. What I instead like to do, is for the initial screen to display “ Welcome to Maze Program. Please log in, or sign up”
* Below, 2 boxes will be present. One of them for the username, and the other for the password. The password should not be visible and all the characters should look like \*.
* So instead of password1, the user would see \*\*\*\*\*\*\*\*\*. There would be a button that lets the user look at the password, which will show the true password the user entered instead of stars.
* Then, there will be a button to “Log in” under the two boxes mentioned. If both the password and the username are correct, the user can enter the program.
* Below “Log in” button, there will be “Sign up” button which will open a new window to let the user sign up to the program, and the new information will be stored in the database.
* To enhance security, the passwords are not directly stored in the database, instead a hashing algorithm will be used. When the user first signs up, the password they enter is hashed, and the hash is stored in a database.
* When the user tries to log in, the password they entered is also hashed, and the hashed password entered is to be compared with the stored hash password for the same username, if they correspond, the user is granted access to the program.

To-do list (12 Nov) - Done

* When I click on “Sign up” it should terminate the Sign Up window.
* In the sign up window, the password is currently visible at all times. I would like the password to not be visible and instead automatically displayed as stars, \*\*\*\*, with the button to toggle visibility between seeing the real password and stars. (The same as the original login page)
* When I enter “admin123” for both the password and the username, instead of logging me into the program, the program needs to open a special window where all the current usernames are displayed. The passwords are not displayed, because the program doesn’t know the password, as I have said the passwords are salted and hashed.
* I also want a scroll bar that allows me to look at the entire list of users incase it is too large, similar to how I have a scroll bar in the my\_mazes menu.
* At the top there should be a button to Quit the program, and a button to “Delete all” usernames, which will ask for confirmation, similar to how the my\_mazes menu works.
* Next to every username , there will be displayed a “Delete” button that will delete the current username and the stored hash of the password corresponding to the username from the database.
* The delete all button should not delete the “admin123” username, Instead of this special case being stored in the database, it should be stored as an if statement for example, if the user enters “admin123” as username and “admin123” as password.
* Also, the user should not be able to sign up an account named “admin123” despite the fact that such account is not in the database.

To-do list (16 Nov) - Done

* Add correct comments throughout the whole program
* For my imports
* For my maze class
* For my UserAuthentication class
* For the MazeApplication class
* For the methods that have the logic for perfect and non-perfect mazes
* For the main execution block

To-do list (22 Dec) - Done

* Add restrictions to not be able to make more than 1 signup window at once
* User should be able to enter alphanumeric characters for password and username only, no greek letters or emojis etc
* Also the password and username have to be 3-16 characters each