Task 0: Explain what you are doing/ going to accomplish

Create the basic quiz program and get it to work on the website

Task 1: Sketch interface design

*Draft a rough design for the interface that allows the user to trigger functionality in task 1, while also annotating where the information in task 2 will be displayed. Create another sketch listing the interface widgets used to create the interface.*

Task 2: Identify any classes required

*Explain what the class will represent, plus listing what information will be stored in the class and any functions the class will have.*

Task 3: Identify information to be displayed

*What information will the interface need to display to the user?*

An image, the question, the choices and a next button

Task 4: Identify user inputs

*What program functions can the user trigger through the interface?*

Task 5: Identify any constants or existing data if required

question\_list = [

question(“cover.png”, “question”, “choice\_1”, “choice\_2”, “choice\_3”, “choice\_4”)

…

…

…

]

Task 6: Identify indexed data structures

Task 7: Determine what calculations are necessary

*Write out the calculations the program will have to compute.*

Task 8: Develop a modular structure for your program

*Describe any functions that the computer program will have, identifying any sub-functions where required.*

From itertools import count

create class called Questions:

\_ids = count(0)

Task 9: Define the functions identified

*Describe the functions for both the main program and any classes in terms of input and/or output where required. You may choose to do this with flow charts or pseudo-code (not Python code!). Add in additional steps or explanations using sequential, conditional, iterative statements where required. Identify global and/or local variables.*

Define function called \_\_init\_\_(self, image, question, choice\_1, choice\_2, choice\_3, choice\_4):

self.id = next(self.\_ids)

self.cover = image

self.question = question

self.choice\_1 = choice\_1

self.choice\_2 = choice\_2

self.choice\_3 = choice\_3

self.choice\_4 = choice\_4

Task 10: Address any relevant implications such as usability, functionality, legal/ethical requirements.

Task 11: Document test cases for testing the program

*Document any testing that can be used to test your program. If any input is inputted using the keyboard, describe the expected input, plus any exceptional, boundary or invalid cases.*

Task 12: Refine the plan

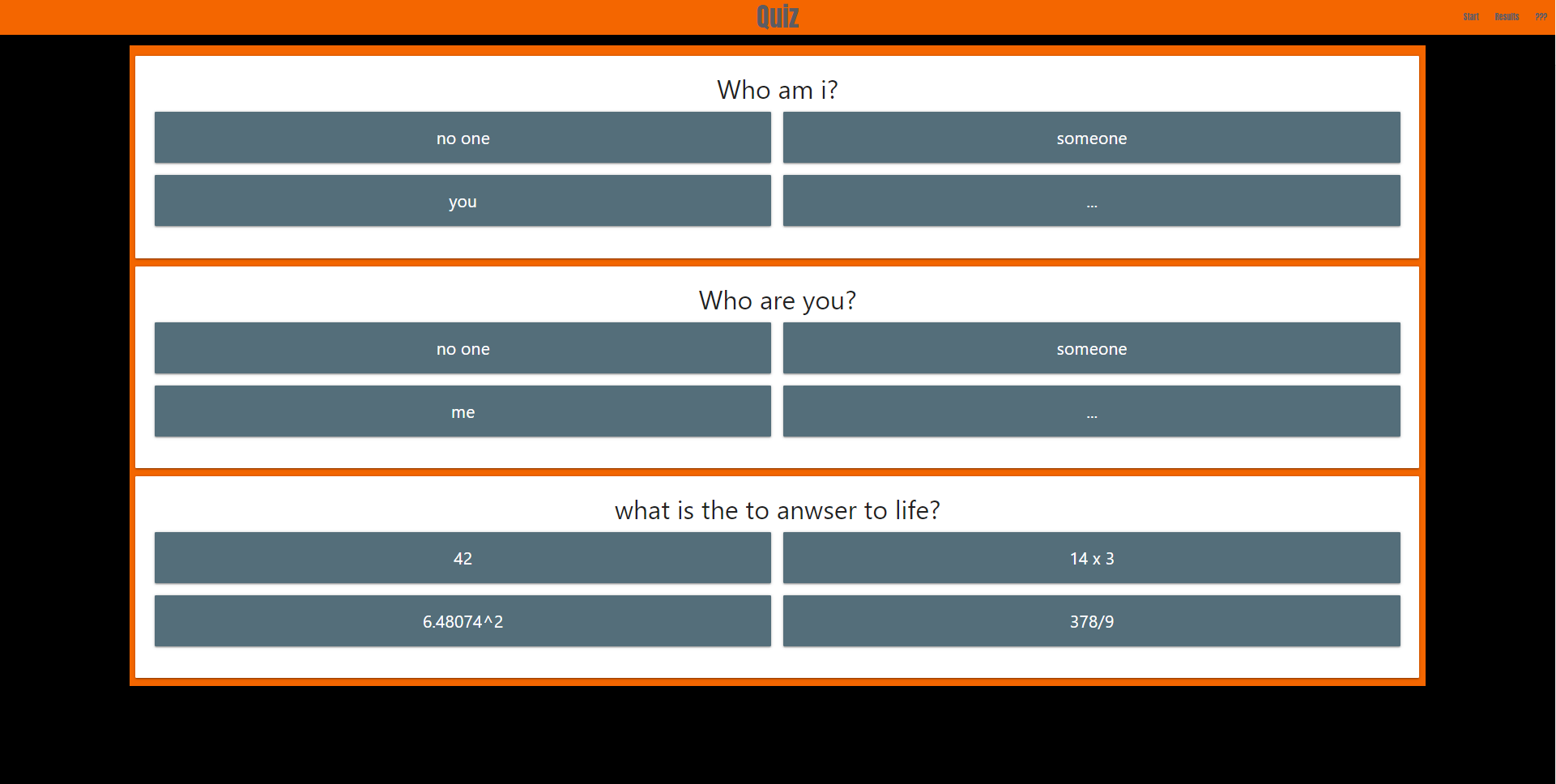
*Note any modifications here when iterating through the development cycles.*

Renamed the python program to ‘Run.py’ to make it easier for me to find it in the folder.

I have added a new variable in the list called ‘state’. This will start with a value of 0, this would change to a 1 or 2 if the chosen choice is correct or incorrect. 0 is not answered.

Task 13: Document testing

*Show screenshots of your program working with descriptions of each image. These images should test the tests cases listed above.*



Task 14 : Evaluation

*How did your version turn out*

I have got it to display the questions and choices for each one, I have left out the selecting the choices and the next button as I think it should be worked out in the next version.