PROJECT REPORT

1. The quiz game can be played by 3 players.
2. To generate questions, I have used a script. The question list is basically a list of the indices and the answer list is identical to the question list.

   (The script I have used is:

   ```
   Q = []
   for i in range (100):
       Q.append(i)
   ```

)

   This method of creating questions was the easiest way to make 100 questions, although they can be substituted with real questions and answers if necessary.

3. The question will be sent by picking a random number within the number of questions and then the program will check if the question has already been sent in which case it will pick a different question.
4. The answer is just the question itself. As the question each of the participants will get a number (the same number) and they have to press the buzzer if they know the answer. Buzzer can be pressed by pressing any key on the keyboard and then pressing enter. They have to buzz within 10 secs or their buzz will not be accepted and the question will be discarded. The same question will not be shown again. After buzzing the word “buzzed” will be displayed. The player now has 10 secs to answer.
5. When the client answers, the server tells the client if their buzzer or their answer is valid and is recorded based on whether it has been sent within 10 secs. **The server will only tell you whether 10 secs have passed or not after you send a message to the it. If the client has answered after 10 secs the question will be discarded.**

6. If the answer is correct their score will increase by one else it will decrease by ½. They game will then move on to the next question and this process will be repeated.
7. If the quiz goes on to complete 50 questions without reaching a winner the game will be declared a tie.

Code Description:

This code is written in python. It used the following modules:

1. Socket
2. Random
3. Sys
4. Thread
5. Select
6. Time

I have maintained some global lists and variables in server.py. I have first created a socket object and bound it to my IP and Port number. I have a list called sent_questions to avoid repeat of questions. I use a buzzer list in which I store the buzzer value (1 if buzzed, 0 if not) and the question number.

I use a client list to keep track of which client I am communicating with now.

The function send_question picks a question and send it to the clients.

The main_fun function is the main algorithm of the program. This function consists of an infinite loop and before every iteration I store the time in seconds. During the iteration if I don't get a message in 10 secs, I print something depending on whether the buzzer has been pressed. If yes than I print the time to answer has been exceeded (and make buzzer [0] = 0), else I will print time to buzz has been exceeded and I will send the next question.

If I do get a message then again, I will check the buzzer value. If 0, then I will make buzzer [0] = 1 and keep track of which client I am communicating with so that I

can make sure that I can change the score of the right client depending on their answer.

Else I check if the message is from the client who buzzed and award them scores based on their answer and then I make buzzer [0], 0 again. I maintain a list called score_counter to keep track of the scores of all the clients, if the client who has answered has score > 5 I declare him the winner and the game ends. Otherwise the next question is sent to the clients.

How to run the project:

Go to the directory in the terminal where the files are stored

Enter the command: python server.py

Note this code will throw an error if you use python3 due to some import issues

Open 3 other terminals go to the directory and enter the command: python client.py.

The quiz has begun.

NOTE: To check the timing condition, wait for 10 seconds before pressing the buzzer or the answer and then enter the message.