Aim: To create a multiple-choice buzzer quiz which supports 3 players and 1 host using python3 and socket programming.

Requirements:

1. The program is written using 'localhost' as the server and port number '1234'.

2. The user needs to have python3 installed on their pc.

3. The user needs to have 'inputimeout' python module installed. The command to install 'inputimeout' module is "pip install inputimeout".

4.The client should not send input as "N".

Assumptions:

1. There are three players.

2. Correct answer is +1 point and wrong answer is -0.5 points.

3. First player to reach 5 points wins.

4.The clients will receive messages only after their input has timed out if they are required to input an answer.

Instructions:

1. Run the server.py file using "python3 server.py".

2. Run the client file 3 times using "python3 client.py".If all 3 clients have been connected to the server then the quiz will start.

3. The clients are designated as 'Player 1','Player 2 ','Player 3' based on the order in which they connected to the server.

4. The server will send questions one by one to the clients.

5. Once the client has received the question a message will be displayed saying "Waiting for Input".The first client to input any key will be given a chance to answer the question.

6. All of the clients will receive another message saying "Player x has pressed the buzzer".

7. The client who has pressed the buzzer first will receive another message saying "Waiting for input". He should then input an answer.

8. Whether he answered correct or wrong will be displayed to all the clients and the quiz will continue.

9. The quiz will end when any client receives 5 points or all the questions have been displayed in which case nobody will win the game.

Description of the Code in the server side:

1.The quiz is implemented using socket programming in python3.

2.We first a create a host who is waiting for 3 clients to connect.

3.We work with the clients using multiple threads.

4.Once all of the clients have been connected the quiz will start.

5.Host sends questions and messages to the clients using the 'broadcast' function. Upon receiving a message from the client, we process it in the client_thread.

6.If the host is waiting for a buzzer then it puts the address of the first client who sent a message on 'first_client' variable and changes check [0] to 1.

7.After the buzzer has been pressed if the client who sends a message is not the first client then we do not process the message and simply reply that another client has pressed the buzzer.

8.After the buzzer has been pressed if the client who sends a message is the first client then we process the message and check whether the answer is correct or not and change their score accordingly in 'scores'.

9.When someone's score crosses 5 points or all of the questions have been exhausted, we call the function 'game over'.

10.If all 3 clients send messages then the server moves onto the next question.

Description of the code on the client side:

1.The client will be continuously printing messages that he receives from the server.

2.Once the client receives the message "1" he calls the 'give reply' function and waits 10 seconds for the user input.

3.If the client doesn't input anything within 10 seconds then the input times out and a message "N" is automatically sent.