**Level 1**

1. Read through
2. Code
   1. print(“Type your name.”)
   2. name=input()
   3. print(“Hi”, name, “how are you?”)
3. Code
   1. print(“Which number is needed to square root?”)
   2. sr=int(input())
   3. Import math
   4. sr-(math.sqrt(sr))
   5. print(sr)

**Level 2**

1. Link <https://repl.it/@Janky_God_Doby/Player-vs-Player-Tic-Tac-Toe>
2. Code in link
3. My program keeps track of my game mode is by having the list titled “Board” which is located on line 4. The list begun with indexes which begins from 0-9. However, I used the indexes 1-9 for my tic tac toe board. I have done this by making a function called “showBoard” in which it places the indexes 1-3 on the top row, 4-6 on the middle row and 7-9 on the bottom row in order to make the area for where the player or the computer place their slots. Nextly, I have also made dark lines which will pass through the board to help formulate it to become the proper tic tac toe board. I had many vertical and horizontal lines to separate each of the rows and columns. These lines are shown as strings and to do this, I had to use the print command print() which is in the showBoard function. The player controls the X term in tic tac toe. For it to move, the human whom is playing X will have to move wherever they desire as long as it’s between (1-9). These indexes will be replaced by X. The computer controls the O term in Tic Tac Toe. For it to move, the number is generated by a random number generator which means that the O which is controlled by the computer will move completely random. If the spaces are already taken by the indexes X or O it will display a message in which you cannot put it in that particular spot. The spots which are open will display regular numbers and the tic tac toe board is updated with every single computer or player move.
4. The player who is (X) is given the choice of placing their index from 1-9 on the grid. Once the player chooses the number, the number is then changed into an integer with the int(input)(). This can be identified as an index of the Board. The number gets checked only if the number is a valid move. Due to this, the program checks if the index is either X or O and if the index is not selected the automatic selected index will automatically become X using the “if” term. If there is a number or letter which is invalid it will display a message in which it tells the player that they should put a valid number. (Line 44-48). I used “continue;” in order to let the player try again if they accidently typed an invalid number or letter. This means that the program will continue to work even if the player has put an invalid number.
5. The player switches between both player X and player O which means the loop will always become true unless it ends with the “break;” command. This means that the moves will continue between both player X and player O until one of the players eventually wins the game. Once the loop, both the player X is asked to move and the player O (computer) moves and this cycle will repeat until one of these players eventually gets 3 in a row. The game automatically stops when the board is completely full due to their being no moves left or when there is a clear winner. After a winner is determined, the loop will stop due to the “break” and if the player (who is X) will receive this indication, they will earn the indication of “("You win player #”). If the computer (O) wins then the player will receive this message, '!!Pathetic, the Computer has beated you, You Lose!!”. (Line 105-this may change due to change of code). The entire program will end either if the player loses, wins or if there are no moves left. In order to keep track of the gameplay for Tic Tac Toe, the function “showBoard” is placed in the loop.

**Level 3**

1. The strategy that was being made for determining if player “X” and player “O” has won the game after a move is made is by having combinations. These combinations mean that if the player or computer will have 3 integers in a row, they will win the game. (Line 31-34) If the player has won the game, they will earn the “!!Player # won!”. (Line 93-98). To determine the draw, I had to implement the code “If checkWinner (‘O’) == False and checkWinner (‘X’) == False” in order to determine a draw. This code means that if both of the players will have not been able to get 3 in a row and filled up the board, it will lead to a draw.
2. On Link above
3. On Link above