Rivan Adhikari
JT Swiderski
Cody Wu
Patrick Simon

1. Create a list of lists (grid) to represent the Go board.
    a. Letters will represent the rows starting with a as row one
    b. Numbers will represent the columns from left to right
2. Print the board (should be only periods).
3. Create a function that tells us if the spot is empty or if it contains another piece already.
4. Make sure black goes first and every turn after alternates between black and white.
    a. Make a loop to count the number of turns and compare the 2 players making player a go if the turns are equal and player 2 goes if player 1 has one more turn.
    b. Add 1 to turn count at end of turn after any possible error message for invalid entry.
5. Ask for input from player for their turn
    a. Display an error message if a piece exists there already
    b. If input is blank, convert to their symbol and replace the period on the board
6. Print the board
7. When the user enters “stop” or all positions are O or o, end the game.

Algorithm:

```
#Creating the grid
For i in range(9)
   Print(i)
   For i in range(9)
      print(i)
a = [0,1,2,3,4,5,6,7,8]
b = [0,1,2,3,4,5,6,7,8]
c = [0,1,2,3,4,5,6,7,8]
d = [0,1,2,3,4,5,6,7,8]
e = [0,1,2,3,4,5,6,7,8]
f = [0,1,2,3,4,5,6,7,8]
g = [0,1,2,3,4,5,6,7,8]
h = [0,1,2,3,4,5,6,7,8]
i = [0,1,2,3,4,5,6,7,8]

board = [a,b,c,d,e,f,g,h,i]
```

Rivan Adhikari
JT Swiderski
Cody Wu
Patrick Simon

Instructions:
This game is meant for 2 players. Player 1 will be black(capital O) and player 2 will be white (o). The board is numbered with columns being numbers 0 (left-most) through 8 (right-most) and rows assigned letters a(top row) through i(bottom row). Player 1 will start the game by entering the coordinates of his desired move. Player 2 will then take his turn and it will alternate. A player cannot enter coordinates to an intersection that is already occupied by either player's symbol. Continue to alternate turns until the board is full or the players enter “stop” to end the game.