Mock Project

Name: Abhigya Parashar
Emp id: 2005580
Mail id: abhigyaparashar22@gmail.com
Location: Bangalore

Introduction:

- Game Description:
  The name of the game is Dots and Boxes. It is a two-player game. It consists of a board with many dots on it. Each dot is placed perpendicular to each other, thus forming a dotted square or rectangle board. The game is played by joining the dots. On clicking on the region in between two dots, they get connected with a line of same colour as the player. The aim of the game is the form compartments of single colour boundaries so as to secure those boxes. The player with higher number of compartments wins.

- Cases of Win and Lose:
  This game must be played with a well thought of strategy. As, explained in the game description, players try to gain as many secure boxes as possible while blocking the other player from doing the same. So, the winning criteria is higher number of secured boxes. The other player then automatically loses. There can also arise a case of a tie incase both players have equal number of secure boxes. The scoring scheme is:
  Win = 1
  Tie = 0.5
  Lose = 0

- Input and Output format:
  Input is the Name of two players
  Output is the game which is to be played.
  A dialogue box opens in which the game is played. The dialogue box shows the result of the match. There is also an option for rematch.
  The users can close the dialogue box to end the game.
  The Database and Leaderboard files keep getting updated in the background.

- Database/Leaderboard Description:
  The database and leaderboard keep getting updated dynamically. The data is not lost once the program is closed i.e when the program is rerun and a new game is started the previously stored database is available and new values get updated to it.
  Database: It contains 6 columns/features.
    Column 1: Details about first player
    Column 2: Details about second player
    Column 3: Result of the match: Win/Lose/Tie
    Column 4: Date of the match
    Column 5: Time of the match
    Column 6: The duration of the match i.e how long the match was played for (in seconds)
  The database excel file contains information and details about every match that is played.
  Leaderboard: It contains 6 columns/features.
    Column 1: Name of the player

Column 2: Cumulative score of the player considering total wins, loses and ties
Column 3: Total number of matched that the player has played
Column 4: Of the total matches played, the number of matches that ended in a win.
Column 5: Shortest time taken by the player in any match that he has won
Column 6: The win percentage of the player

The leaderboard considering these features sorts the players in the priority of win percentage, score, number of matches played, Shortest time taken to win.

Code Snippet:

```
from tkinter import *
import numpy as np
import pandas as pd
from openpyxl import load_workbook

size_of_board = 400
number_of_dots = 5
symbol_size = (size_of_board / 3 - size_of_board / 8) / 2
symbol_thickness = 50
dot_color = '#7BC043'
player1_color = '#0492CF'
player1_color_light = '#67B0CF'
player2_color = '#EE4035'
player2_color_light = '#EE7E77'
Green_color = '#7BC043'
dot_width = 0.25*size_of_board/number_of_dots
edge_width = 0.1*size_of_board/number_of_dots
distance_between_dots = size_of_board / (number_of_dots)
res=0

class Dots_and_Boxes():

    # ------------------------------------------------------------------
    # Initialization functions
    # ------------------------------------------------------------------
    def __init__(self):
        self.window = Tk()
        self.window.title('Dots_and_Boxes')
        self.canvas = Canvas(self.window, width=size_of_board,
height=size_of_board)
        self.canvas.pack()
        self.window.bind('<Button-1>', self.click)
        self.player1_starts = True
        self.refresh_board()
        self.play_again()
        self.df = pd.read_excel('stats.xlsx',index_col=0)
        if self.df.empty:
            self.df = pd.DataFrame({'Player 1': [], 'Player 2': [],'Res':
[],'Date':[],'Time':[],'Timediff':[]})
        #self.lb= pd.read_excel('leaderboard.xlsx',index_col=0)
        #if self.lb.empty:
        self.lb = pd.DataFrame({'Name': [], 'Score': [],'Match no':
[],'Wins':[],'Shortest Time':[],'Win%':[]})


    def append_dataset(self,x,y,z,d,t,tdiff):
        temp_df=pd.DataFrame({'Player 1': [x], 'Player 2': [y],'Res':
[z],'Date':[d],'Time':[t],'Timediff':[float(tdiff)]})
        self.df=self.df.append(temp_df,sort=False)
        self.df.to_excel('stats.xlsx')
```

```
    def leaderboard(self):
        self.df = pd.read_excel('stats.xlsx',index_col=0)
        name=self.df['Player 1'].tolist()
        names=self.df['Player 2'].tolist()
        for i in range(len(self.df['Player 1'])):
            if name[i] not in self.lb['Name'].tolist():
                temp_lb=pd.DataFrame({'Name':[name[i]],'Score':[0],'Match
no':[1],'Wins':[0],'Shortest Time':[999],'Win%':[0]})
                self.lb=self.lb.append(temp_lb,sort=False)
            else:
                self.lb.loc[self.lb['Name']==name[i],'Match no']+=1
            if self.df['Res'].values[i]==1:
                self.lb.loc[self.lb['Name']==name[i],'Score']+=1
                self.lb.loc[self.lb['Name']==name[i],'Wins']+=1
                if
float(self.df['Timediff'].values[i])<float(self.lb.loc[self.lb['Name']==name[i],
'Shortest Time']):
                    self.lb.loc[self.lb['Name']==name[i],'Shortest
Time']=self.df['Timediff'].values[i]
            elif self.df['Res'].values[i]==0:
                self.lb.loc[self.lb['Name']==name[i],'Score']+=0.5

self.lb.loc[self.lb['Name']==name[i],'Win%']=float(self.lb.loc[self.lb['Name']==
name[i],'Wins'])/float(self.lb.loc[self.lb['Name']==name[i],'Match no'])*100

        for i in range(len(self.df['Player 2'])):
            if names[i] not in self.lb['Name'].tolist():
                temp_lb=pd.DataFrame({'Name':[names[i]],'Score':[0],'Match
no':[1],'Wins':[0],'Shortest Time':[999],'Win%':[0]})
                self.lb=self.lb.append(temp_lb,sort=False)
            else:
                self.lb.loc[self.lb['Name']==names[i],'Match no']+=1
            if self.df['Res'].values[i]==2:
                self.lb.loc[self.lb['Name']==names[i],'Score']+=1
                self.lb.loc[self.lb['Name']==names[i],'Wins']+=1
                if
float(self.df['Timediff'].values[i])<float(self.lb.loc[self.lb['Name']==names[i]
,'Shortest Time']):
                    self.lb.loc[self.lb['Name']==names[i],'Shortest
Time']=self.df['Timediff'].values[i]
            elif self.df['Res'].values[i]==0:
                self.lb.loc[self.lb['Name']==names[i],'Score']+=0.5

self.lb.loc[self.lb['Name']==names[i],'Win%']=float(self.lb.loc[self.lb['Name']=
=names[i],'Wins'])/float(self.lb.loc[self.lb['Name']==names[i],'Match no'])*100

        self.lb.sort_values(by=['Win%','Score','Match no','Shortest
Time'],inplace=True,ascending=(False,False,False,True))
        #print(self.lb)
        self.lb.to_excel('leaderboard.xlsx')

    def play_again(self):
        self.refresh_board()
        self.board_status = np.zeros(shape=(number_of_dots - 1, number_of_dots -
1))
        self.row_status = np.zeros(shape=(number_of_dots, number_of_dots - 1))
        self.col_status = np.zeros(shape=(number_of_dots - 1, number_of_dots))

        # Input from user in form of clicks
        self.player1_starts = not self.player1_starts
        self.player1_turn = not self.player1_starts
        self.reset_board = False
        self.turntext_handle = []

        self.already_marked_boxes = []
```

```
        self.display_turn_text()

    def mainloop(self):
        self.window.mainloop()

    # ------------------------------------------------------------------
    # Logical Functions:
    # The modules required to carry out game logic
    # ------------------------------------------------------------------

    def is_grid_occupied(self, logical_position, type):
        r = logical_position[0]
        c = logical_position[1]
        occupied = True

        if type == 'row' and self.row_status[c][r] == 0:
            occupied = False
        if type == 'col' and self.col_status[c][r] == 0:
            occupied = False

        return occupied

    def convert_grid_to_logical_position(self, grid_position):
        grid_position = np.array(grid_position)
        position = (grid_position-
distance_between_dots/4)//(distance_between_dots/2)

        type = False
        logical_position = []
        if position[1] % 2 == 0 and (position[0] - 1) % 2 == 0:
            r = int((position[0]-1)//2)
            c = int(position[1]//2)
            logical_position = [r, c]
            type = 'row'
            # self.row_status[c][r]=1
        elif position[0] % 2 == 0 and (position[1] - 1) % 2 == 0:
            c = int((position[1] - 1) // 2)
            r = int(position[0] // 2)
            logical_position = [r, c]
            type = 'col'

        return logical_position, type

    def mark_box(self):
        boxes = np.argwhere(self.board_status == -4)
        for box in boxes:
            if list(box) not in self.already_marked_boxes and list(box) !=[]:
                self.already_marked_boxes.append(list(box))
                color = player1_color_light
                self.shade_box(box, color)

        boxes = np.argwhere(self.board_status == 4)
        for box in boxes:
            if list(box) not in self.already_marked_boxes and list(box) !=[]:
                self.already_marked_boxes.append(list(box))
                color = player2_color_light
                self.shade_box(box, color)

    def update_board(self, type, logical_position):
        r = logical_position[0]
        c = logical_position[1]
        val = 1
        if self.player1_turn:
            val =- 1
```

```
        if c < (number_of_dots-1) and r < (number_of_dots-1):
            self.board_status[c][r] += val

        if type == 'row':
            self.row_status[c][r] = 1
            if c >= 1:
                self.board_status[c-1][r] += val

        elif type == 'col':
            self.col_status[c][r] = 1
            if r >= 1:
                self.board_status[c][r-1] += val

    def is_gameover(self):
        return (self.row_status == 1).all() and (self.col_status == 1).all()

    # -----------------------------------------------------------------
    # Drawing Functions:
    # The modules required to draw required game based object on canvas
    # -----------------------------------------------------------------

    def make_edge(self, type, logical_position):
        if type == 'row':
            start_x = distance_between_dots/2 +
logical_position[0]*distance_between_dots
            end_x = start_x+distance_between_dots
            start_y = distance_between_dots/2 +
logical_position[1]*distance_between_dots
            end_y = start_y
        elif type == 'col':
            start_y = distance_between_dots / 2 + logical_position[1] *
distance_between_dots
            end_y = start_y + distance_between_dots
            start_x = distance_between_dots / 2 + logical_position[0] *
distance_between_dots
            end_x = start_x

        if self.player1_turn:
            color = player1_color
        else:
            color = player2_color
        self.canvas.create_line(start_x, start_y, end_x, end_y, fill=color,
width=edge_width)

    def display_gameover(self):
        player1_score = len(np.argwhere(self.board_status == -4))
        player2_score = len(np.argwhere(self.board_status == 4))

        if player1_score > player2_score:
            # Player 1 wins
            text = 'Winner: ' + Player1
            color = player1_color
            res=1
        elif player2_score > player1_score:
            text = 'Winner:' + Player2
            color = player2_color
            res=2
        else:
            text = 'Its a tie'
            color = 'gray'
            res=0
        x=pd.datetime.now()
        d=str(x.day)+'/'+str(x.month)+'/'+str(x.year)
        t=str(x.hour)+':'+str(x.minute)
        tdiff=(x-d1).seconds
```

```
        self.append_dataset(Player1,Player2,res,d,t,tdiff)

        self.canvas.delete("all")
        self.canvas.create_text(size_of_board / 2, size_of_board / 3, font="cmr
60 bold", fill=color, text=text)

        score_text = 'Scores \n'
        self.canvas.create_text(size_of_board / 2, 5 * size_of_board / 8,
font="cmr 40 bold", fill=Green_color,
                                text=score_text)

        score_text = Player1 + ': '  + str(player1_score) + '\n'
        score_text += Player2 + ': '  + str(player2_score) + '\n'
        # score_text += 'Tie                    : ' + str(self.tie_score)
        self.canvas.create_text(size_of_board / 2, 3 * size_of_board / 4,
font="cmr 30 bold", fill=Green_color,
                                text=score_text)
        self.reset_board = True

        score_text = 'Click to play again \n'
        self.canvas.create_text(size_of_board / 2, 15 * size_of_board / 16,
font="cmr 20 bold", fill="gray",
                                text=score_text)

    def refresh_board(self):
        for i in range(number_of_dots):
            x = i*distance_between_dots+distance_between_dots/2
            self.canvas.create_line(x, distance_between_dots/2, x,
                                    size_of_board-distance_between_dots/2,
                                    fill='gray', dash = (2, 2))
            self.canvas.create_line(distance_between_dots/2, x,
                                    size_of_board-distance_between_dots/2, x,
                                    fill='gray', dash=(2, 2))

        for i in range(number_of_dots):
            for j in range(number_of_dots):
                start_x = i*distance_between_dots+distance_between_dots/2
                end_x = j*distance_between_dots+distance_between_dots/2
                self.canvas.create_oval(start_x-dot_width/2, end_x-dot_width/2,
start_x+dot_width/2,
                                        end_x+dot_width/2, fill=dot_color,
                                        outline=dot_color)

    def display_turn_text(self):
        text = 'Next turn: '
        if self.player1_turn:
            text += Player1
            color = player1_color
        else:
            text += Player2
            color = player2_color

        self.canvas.delete(self.turntext_handle)
        self.turntext_handle = self.canvas.create_text(size_of_board -
5*len(text),
                                                       size_of_board-
distance_between_dots/8,
                                                       font="cmr 15 bold",
text=text, fill=color)


    def shade_box(self, box, color):
        start_x = distance_between_dots / 2 + box[1] * distance_between_dots +
edge_width/2
```

```
        start_y = distance_between_dots / 2 + box[0] * distance_between_dots +
edge_width/2
        end_x = start_x + distance_between_dots - edge_width
        end_y = start_y + distance_between_dots - edge_width
        self.canvas.create_rectangle(start_x, start_y, end_x, end_y, fill=color,
outline='')

    def display_turn_text(self):
        text = 'Next turn: '
        if self.player1_turn:
            text += Player1
            color = player1_color
        else:
            text += Player2
            color = player2_color

        self.canvas.delete(self.turntext_handle)
        self.turntext_handle = self.canvas.create_text(size_of_board -
5*len(text),
                                                       size_of_board-
distance_between_dots/8,
                                                       font="cmr 15
bold",text=text, fill=color)

    def click(self, event):
        if not self.reset_board:
            grid_position = [event.x, event.y]
            logical_positon, valid_input =
self.convert_grid_to_logical_position(grid_position)
            if valid_input and not self.is_grid_occupied(logical_positon,
valid_input):
                self.update_board(valid_input, logical_positon)
                self.make_edge(valid_input, logical_positon)
                self.mark_box()
                self.refresh_board()
                self.player1_turn = not self.player1_turn

                if self.is_gameover():
                    # self.canvas.delete("all")
                    self.display_gameover()
                else:
                    self.display_turn_text()
        else:
            self.canvas.delete("all")
            self.play_again()
            self.reset_board = False


print("The Dots and Boxes Game")
#player_deet()
Player1 = input("Enter the names of the first player: ")
Player2 = input("Enter the names of the second player: ")
print("Switch to different window to play the game")
d1=pd.datetime.now()
game_instance = Dots_and_Boxes()
#game_instance.leaderboard()
game_instance.mainloop()
game_instance.leaderboard()
```

UI Artifacts:

Iteration-1

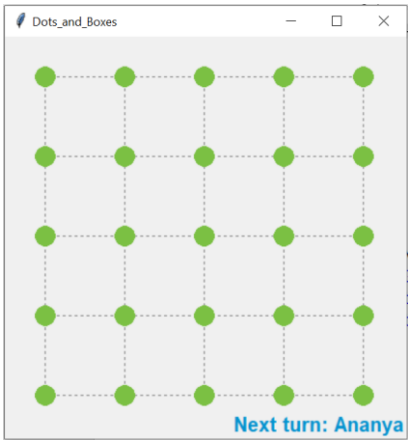
Input:

User input: Names of players

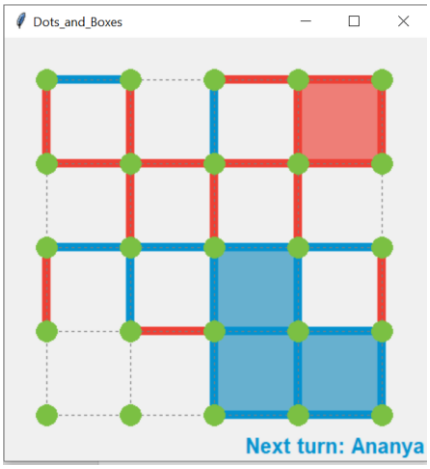
```
= RESTART: C:\Users\Abhigya Parashar\Desktop\
The Dots and Boxes Game
Enter the names of the first player: Ananya
Enter the names of the second player: Abhigya
Switch to different window to play the game
```

Output:

Starting dialogue box with an empty board



The board after playing for a while:



Database:

This contains all details about the match as explained in Game description

	Player 1	Player 2	Res	Date	Time	Timediff
0	Ananya	Abhigya	1	23/10/202	10:29	316

Leaderboard:

Ranking of the players

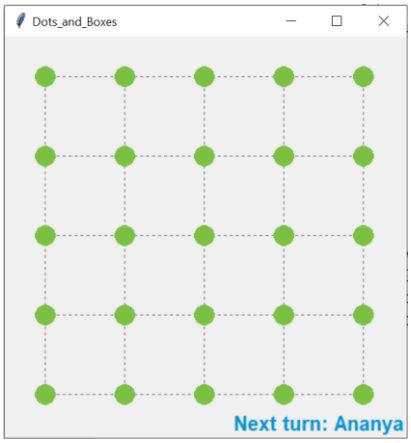
	Name	Score	Match no	Wins	hortest Tim	Win%
0	Ananya	1	1	1	316	100

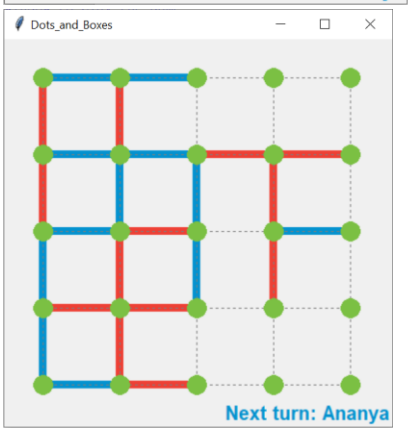
Iteration-2

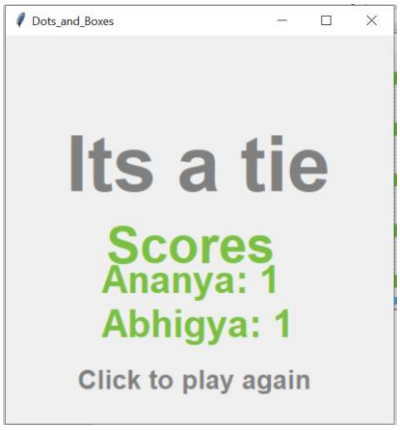
Input:

```
= RESTART: C:\Users\Abhigya Parashar\Desktop\
The Dots and Boxes Game
Enter the names of the first player: Ananya
Enter the names of the second player: Abhigya
Switch to different window to play the game
```

Output:







Updated Database:

	A	B	C	D	E	F	G
		Player 1	Player 2	Res	Date	Time	Timediff
0	0	Ananya	Abhigya	1	23/10/202	10:29	316
0	0	Ananya	Abhigya	0	23/10/202	10:33	510

Updated Leaderboard

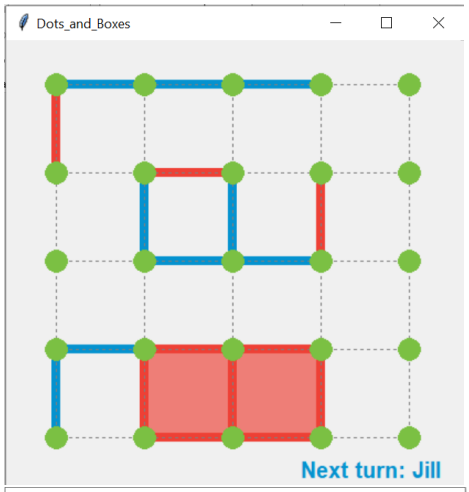
	A	B	C	D	E	F	G
1		Name	Score	Match no	Wins	hortest Tim	Win%
2	0	Ananya	1.5	2	1	316	50
3	0	Abhigya	0.5	2	0	999	0
4							

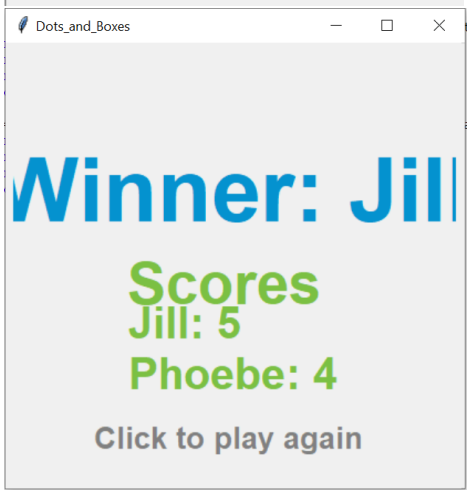
Iteration-3

Input

```
The Dots and Boxes Game
Enter the names of the first player: Jill
Enter the names of the second player: Phoebe
Switch to different window to play the game
```

Output





Updated Database:

	B	C	D	E	F	G
	Player 1	Player 2	Res	Date	Time	Timediff
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124

Updated Leaderboard

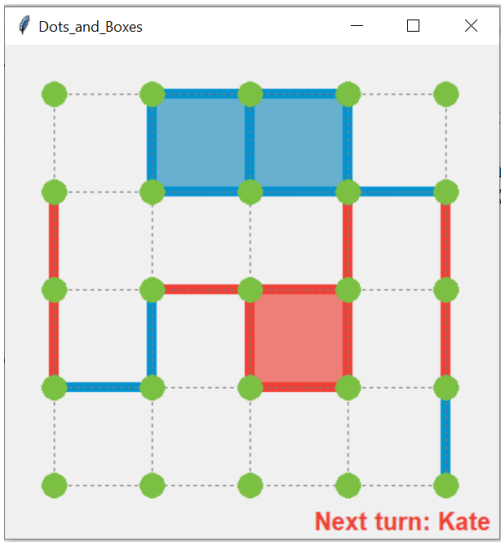
	B	C	D	E	F	G	H
	Name	Score	Match no	Wins	Shortest Time	Win%	
	Jill	1	1	1	124	100	
	Ananya	1.5	2	1	316	50	
	Abhigya	0.5	2	0	999	0	
	Phoebe	0	1	0	999	0	

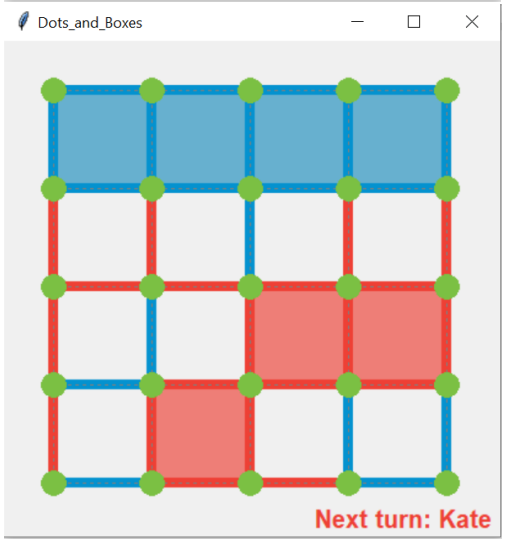
Iteration-4

Input

```
================================== RESTART: C:\Users
The Dots and Boxes Game
Enter the names of the first player: Ananya
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output





Final Updated Database:

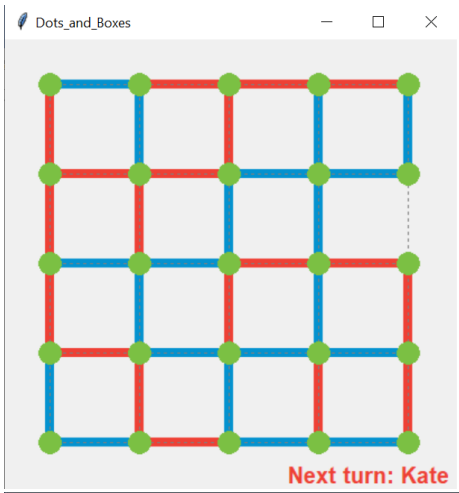
	B	C	D	E	F	G
	Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15	130

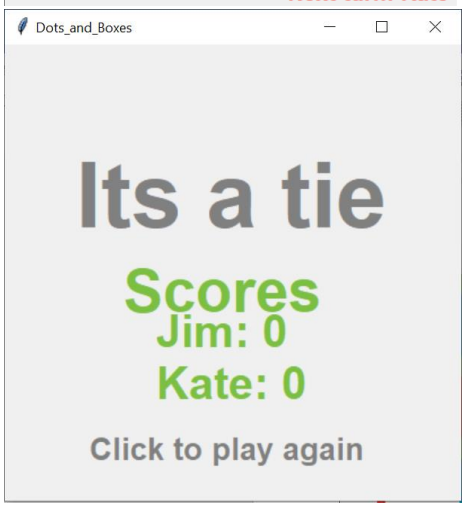
Iteration-5

Input

```
The Dots and Boxes Game
Enter the names of the first player: Jim
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output:





Updated database:

	B	C	D	E	F	G
	Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
	Ananya	Abhigya	1	23/10/2020	10:29	316
	Ananya	Abhigya	0	23/10/2020	10:33	510
	Jill	Phoebe	1	23/10/2020	11:27	124
	Ananya	Kate	1	23/10/2020	12:15:00	130
	Jim	Kate	0	23/10/2020	12:20	231

Updated Leaderboard:

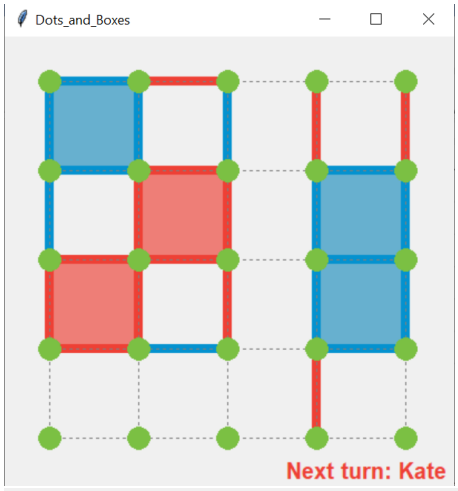
	B	C	D	E	F	G
	Name	Score	Match no	Wins	Shortest Time	Win%
	Jill	1	1	1	124	100
	Ananya	2.5	3	2	130	66.66667
	Abhigya	0.5	2	0	999	0
	Kate	0.5	2	0	999	0
	Jim	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

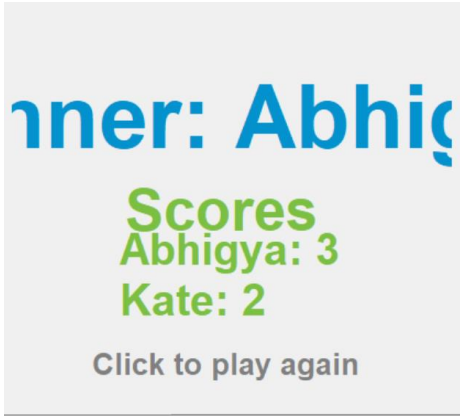
Iteration-6

Input

```
>>>
========== RESTART: C:\
The Dots and Boxes Game
Enter the names of the first player: Abhigya
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output





Updated Database:

	B	C	D	E	F	G
	Player 1	Player 2	Res	Date	Time	Timediff
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205

Updated Leaderboard:

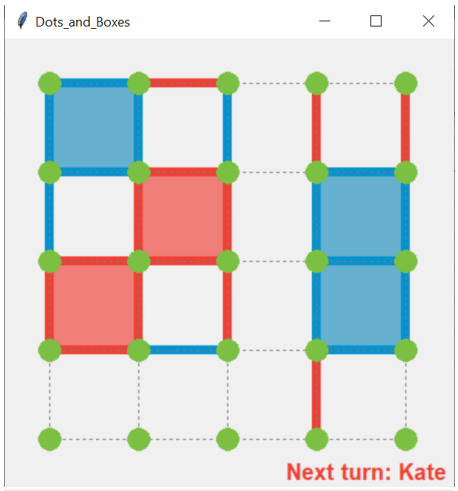
	B	C	D	E	F	G
	Name	Score	Match no	Wins	Shortest Time	Win%
	Jill	1	1	1	124	100
	Ananya	2.5	3	2	130	66.66667
	Abhigya	2.5	3	1	205	33.333
	Kate	0.5	3	0	999	0
	Jim	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

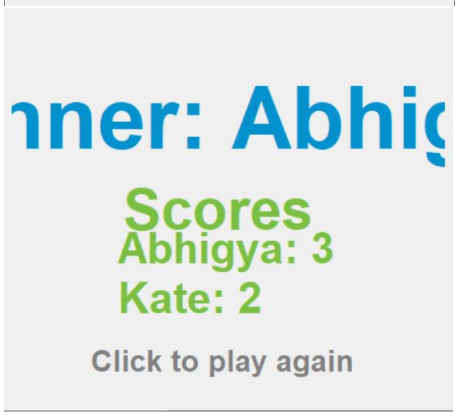
Iteration-7

Input

```
>>>
========== RESTART: C:\
The Dots and Boxes Game
Enter the names of the first player: Abhigya
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output





Updated Database:

B	C	D	E	F	G
Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
Ananya	Abhigya	1	23/10/202	10:29	316
Ananya	Abhigya	0	23/10/202	10:33	510
Jill	Phoebe	1	23/10/202	11:27	124
Ananya	Kate	1	23/10/202	12:15:00	130
Jim	Kate	0	23/10/202	12:20	231
Abhigya	Kate	1	23/10/202	12:26	205
Abhigya	Kate	1	23/10/202	12:30	395

Updated Leaderboard:

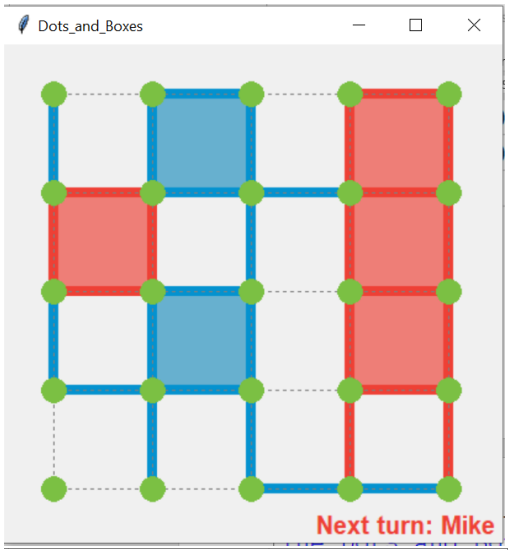
B	C	D	E	F	G
Name	**Score**	**Match no**	**Wins**	**Shortest Time**	**Win%**
Jill	1	1	1	124	100
Ananya	2.5	3	2	130	66.66667
Abhigya	2.5	4	2	205	50
Kate	0.5	4	0	999	0
Jim	0.5	1	0	999	0
Phoebe	0	1	0	999	0

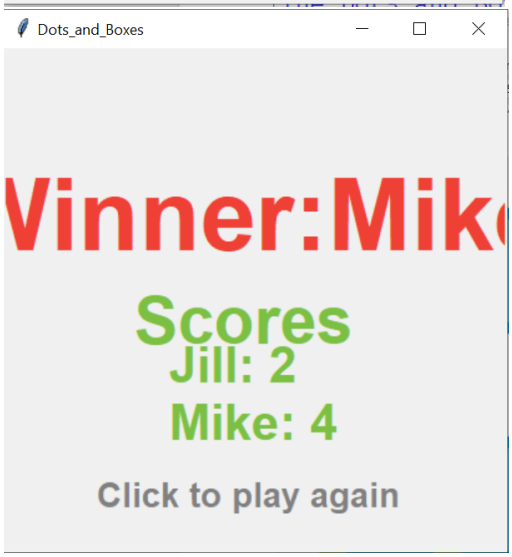
Iteration: 8

Input

```
The Dots and Boxes Game
Enter the names of the first player: Jill
Enter the names of the second player: Mike
Switch to different window to play the game
```

Output





Updated Database:

	B	C	D	E	F	G
	Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260

Updated Leaderboard

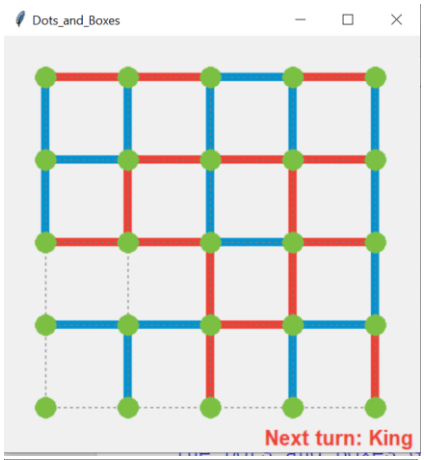
	B	C	D	E	F	G
	Name	**Score**	**Match no**	**Wins**	**Shortest Time**	**Win%**
	Mike	1	1	1	260	100
	Ananya	2.5	3	2	130	66.66667
	Abhigya	2.5	4	2	205	50
	Jill	1	2	1	124	50
	Kate	0.5	4	0	999	0
	Jim	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

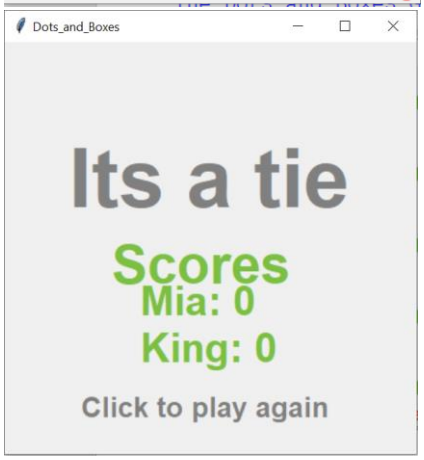
Iteration: 9

Input

```
The Dots and Boxes Game
Enter the names of the first player: Mia
Enter the names of the second player: King
Switch to different window to play the game
```

Output





Updated database:

	B	C	D	E	F	G
	Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260
	Mia	King	0	23/10/202	12:45	120

Updated leaderboard:

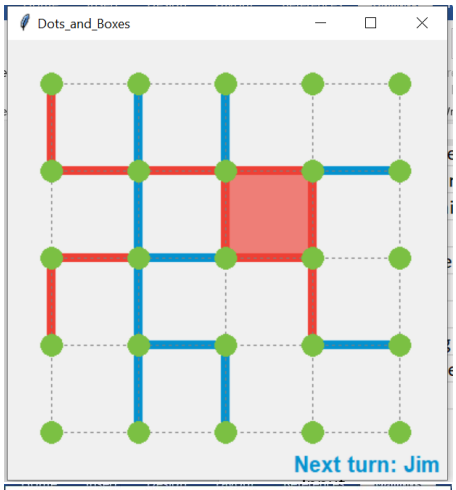
	B	C	D	E	F	G
	Name	**Score**	**Match no**	**Wins**	**Shortest Time**	**Win%**
	Mike	1	1	1	260	100
	Ananya	2.5	3	2	130	66.66667
	Abhigya	2.5	4	2	205	50
	Jill	1	2	1	124	50
	Kate	0.5	4	0	999	0
	Jim	0.5	1	0	999	0
	Mia	0.5	1	0	999	0
	King	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

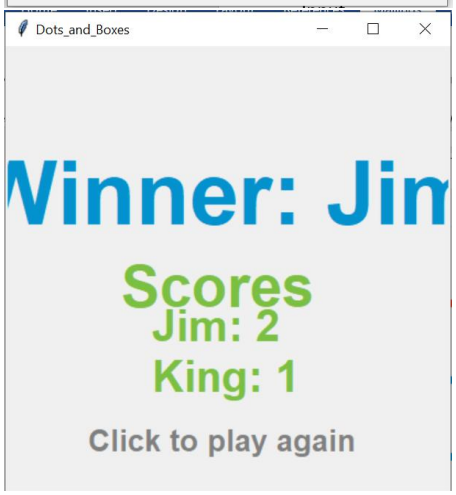
Iteration-10

Input

```
================================== RESTART: 
The Dots and Boxes Game
Enter the names of the first player: Jim
Enter the names of the second player: King
Switch to different window to play the game
```

Output





Updated database:

	B	C	D	E	F	G
	Player 1	**Player 2**	**Res**	**Date**	**Time**	**Timediff**
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260
	Mia	King	0	23/10/202	12:45	120
	Jim	King	1	23/10/202	12:54	276

Updated leaderboard

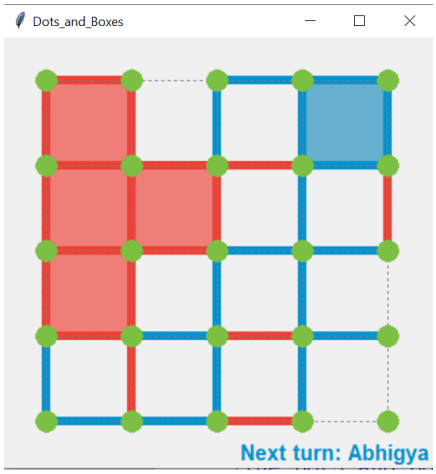
	B	C	D	E	F	G
	Name	**Score**	**Match no**	**Wins**	**hortest Tin**	**Win%**
	Mike	1	1	1	260	100
	Ananya	2.5	3	2	130	66.66667
	Abhigya	2.5	4	2	205	50
	Jim	1.5	2	1	276	50
	Jill	1	2	1	124	50
	Kate	0.5	4	0	999	0
	King	0.5	2	0	999	0
	Mia	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

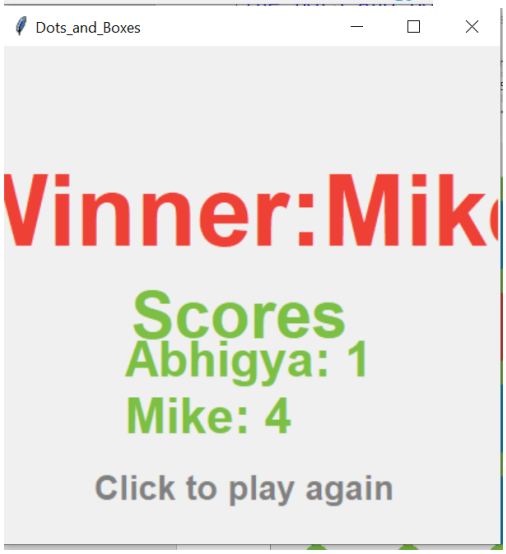
Iteration-11

Input

```
The Dots and Boxes Game
Enter the names of the first player: Abhigya
Enter the names of the second player: Mike
Switch to different window to play the game
```

Output





Updated database

Player 1	Player 2	Res	Date	Time	Timediff
Ananya	Abhigya	1	23/10/202	10:29	316
Ananya	Abhigya	0	23/10/202	10:33	510
Jill	Phoebe	1	23/10/202	11:27	124
Ananya	Kate	1	23/10/202	12:15:00	130
Jim	Kate	0	23/10/202	12:20	231
Abhigya	Kate	1	23/10/202	12:26	205
Abhigya	Kate	1	23/10/202	12:30	395
Jill	Mike	2	23/10/202	12:39	260
Mia	King	0	23/10/202	12:45	120
Jim	King	1	23/10/202	12:54	276
Abhigya	Mike	2	23/10/202	13:1	190

Updated leaderboard

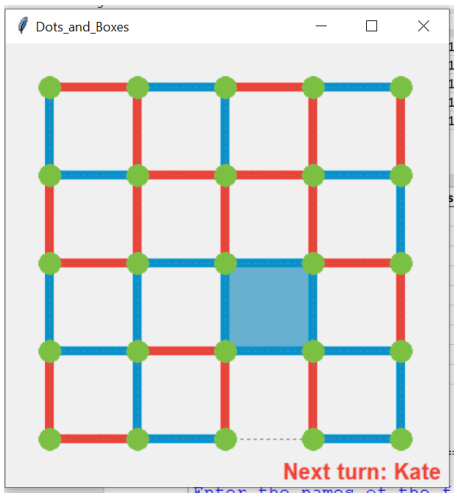
Name	Score	Match no	Wins	Shortest Time	Win%
Mike	2	2	2	190	100
Ananya	2.5	3	2	130	66.66667
Jim	1.5	2	1	276	50
Jill	1	2	1	124	50
Abhigya	2.5	5	2	205	40
Kate	0.5	4	0	999	0
King	0.5	2	0	999	0
Mia	0.5	1	0	999	0
Phoebe	0	1	0	999	0

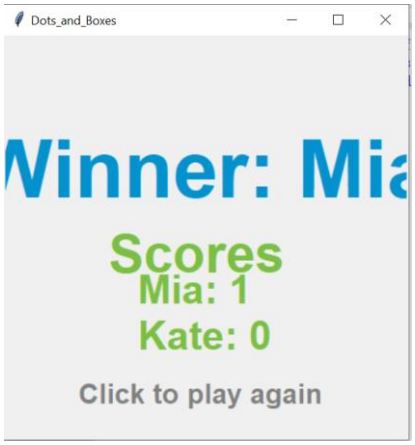
Iteration-12

Input

```
========== RESTART: C:
The Dots and Boxes Game
Enter the names of the first player: Mia
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output:





Updated database:

Player 1	Player 2	Res	Date	Time	Timediff
Ananya	Abhigya	1	23/10/202	10:29	316
Ananya	Abhigya	0	23/10/202	10:33	510
Jill	Phoebe	1	23/10/202	11:27	124
Ananya	Kate	1	23/10/202	12:15:00	130
Jim	Kate	0	23/10/202	12:20	231
Abhigya	Kate	1	23/10/202	12:26	205
Abhigya	Kate	1	23/10/202	12:30	395
Jill	Mike	2	23/10/202	12:39	260
Mia	King	0	23/10/202	12:45	120
Jim	King	1	23/10/202	12:54	276
Abhigya	Mike	2	23/10/202	13:1	190
Mia	Kate	1	23/10/202	13:35	155

Updated leaderboard:

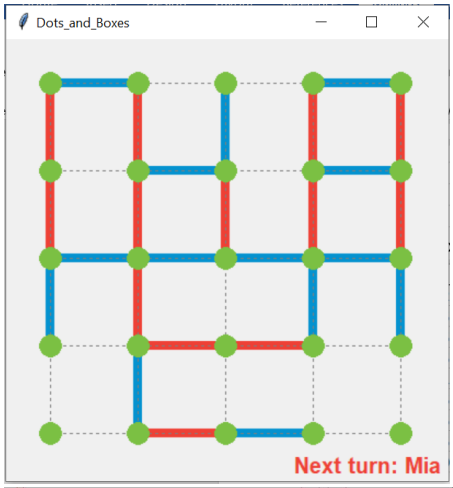
Name	Score	Match no	Wins	hortest Tim	Win%
Mike	2	2	2	190	100
Ananya	2.5	3	2	130	66.66667
Mia	1.5	2	1	155	50
Jim	1.5	2	1	276	50
Jill	1	2	1	124	50
Abhigya	2.5	5	2	205	40
Kate	0.5	5	0	999	0
King	0.5	2	0	999	0
Phoebe	0	1	0	999	0

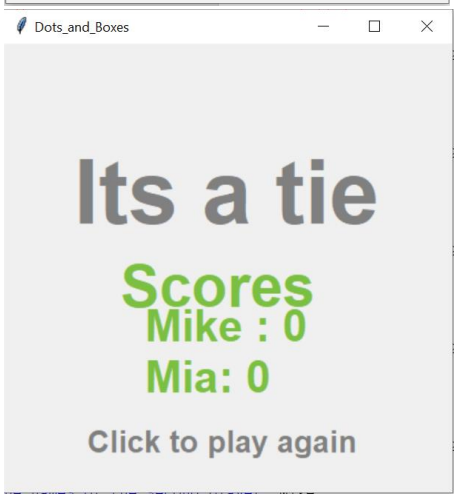
Iteration-13

Input

```
========== RESTART:
The Dots and Boxes Game
Enter the names of the first player: Mike
Enter the names of the second player: Mia
Switch to different window to play the game
```

Output:





Updated database:

	B	C	D	E	F	G
	Player 1	Player 2	Res	Date	Time	Timediff
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260
	Mia	King	0	23/10/202	12:45	120
	Jim	King	1	23/10/202	12:54	276
	Abhigya	Mike	2	23/10/202	13:1	190
	Mia	Kate	1	23/10/202	13:35	155
	Mike	Mia	0	23/10/202	13:41	133

Updated leaderboard:

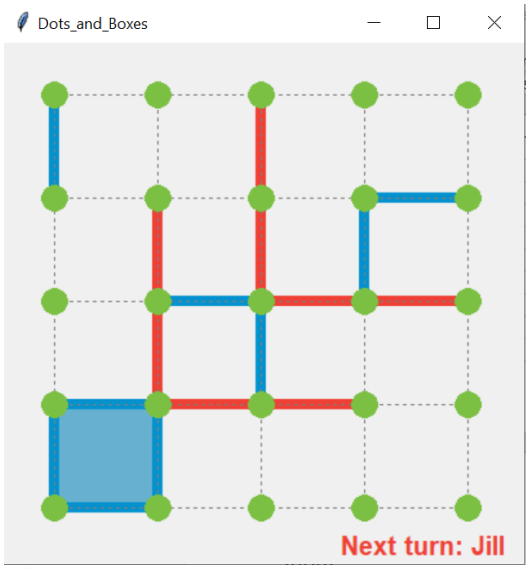
	B	C	D	E	F	G
	Name	Score	Match no	Wins	hortest Tin	Win%
	Mike	2	2	2	190	100
	Ananya	2.5	3	2	130	66.66667
	Jim	1.5	2	1	276	50
	Jill	1	2	1	124	50
	Abhigya	2.5	5	2	205	40
	Mia	2	3	1	155	33.33333
	Kate	0.5	5	0	999	0
	King	0.5	2	0	999	0
	Mike	0.5	1	0	999	0
	Phoebe	0	1	0	999	0

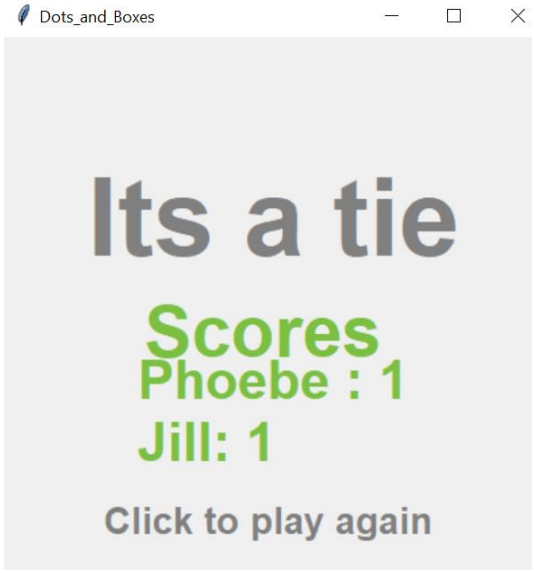
Iteration -14

Input

```
================================== RESTART: C:
The Dots and Boxes Game
Enter the names of the first player: Phoebe
Enter the names of the second player: Jill
Switch to different window to play the game
```

Output





Updated database:

Player 1	Player 2	Res	Date	Time	Timediff
Ananya	Abhigya	1	23/10/202	10:29	316
Ananya	Abhigya	0	23/10/202	10:33	510
Jill	Phoebe	1	23/10/202	11:27	124
Ananya	Kate	1	23/10/202	12:15:00	130
Jim	Kate	0	23/10/202	12:20	231
Abhigya	Kate	1	23/10/202	12:26	205
Abhigya	Kate	1	23/10/202	12:30	395
Jill	Mike	2	23/10/202	12:39	260
Mia	King	0	23/10/202	12:45	120
Jim	King	1	23/10/202	12:54	276
Abhigya	Mike	2	23/10/202	13:1	190
Mia	Kate	1	23/10/202	13:35	155
Mike	Mia	0	23/10/202	13:41	133
Phoebe	Jill	0	23/10/202	13:48	140

Updated leaderboard:

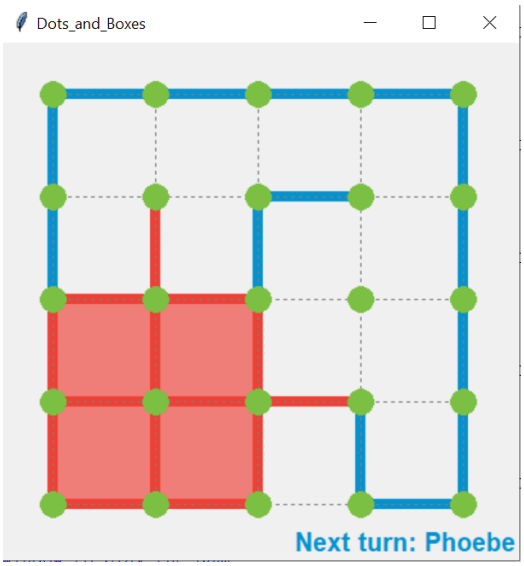
	B	C	D	E	F	G
	Name	Score	Match no	Wins	hortest Tim	Win%
	Mike	2	2	2	190	100
	Ananya	2.5	3	2	130	66.66667
	Jim	1.5	2	1	276	50
	Abhigya	2.5	5	2	205	40
	Mia	2	3	1	155	33.33333
	Jill	1.5	3	1	124	33.33333
	Kate	0.5	5	0	999	0
	King	0.5	2	0	999	0
	Mike	0.5	1	0	999	0
	Phoebe	0.5	2	0	999	0

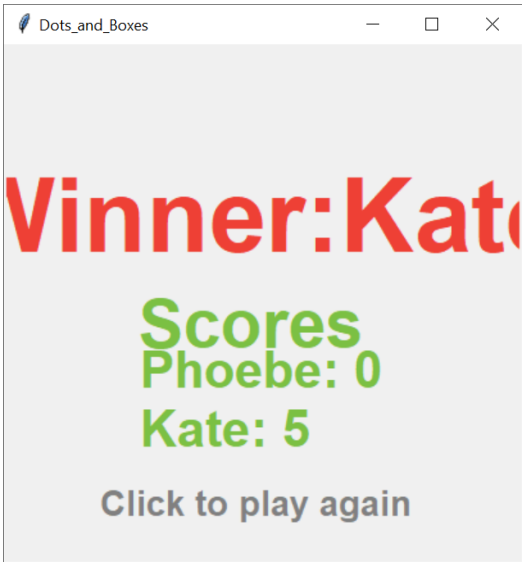
Iteration 15

Input

```
---------------------------------- RESTART: C:
The Dots and Boxes Game
Enter the names of the first player: Phoebe
Enter the names of the second player: Kate
Switch to different window to play the game
```

Output





Updated database:

	Player 1	Player 2	Res	Date	Time	Timediff
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260
	Mia	King	0	23/10/202	12:45	120
	Jim	King	1	23/10/202	12:54	276
	Abhigya	Mike	2	23/10/202	13:1	190
	Mia	Kate	1	23/10/202	13:35	155
	Mike	Mia	0	23/10/202	13:41	133
	Phoebe	Jill	0	23/10/202	13:48	140
	Phoebe	Kate	2	23/10/202	14:46	165

Updated leaderboard:

	B	C	D	E	F	G
	Name	Score	Match no	Wins	hortest Tim	Win%
	Mike	2	2	2	190	100
	Ananya	2.5	3	2	130	66.66667
	Jim	1.5	2	1	276	50
	Abhigya	2.5	5	2	205	40
	Mia	2	3	1	155	33.33333
	Jill	1.5	3	1	124	33.33333
	Kate	1.5	6	1	165	16.66667
	Phoebe	0.5	3	0	999	0
	King	0.5	2	0	999	0
	Mike	0.5	1	0	999	0

Final Database after **15** iterations:

	Player 1	Player 2	Res	Date	Time	Timediff
	Ananya	Abhigya	1	23/10/202	10:29	316
	Ananya	Abhigya	0	23/10/202	10:33	510
	Jill	Phoebe	1	23/10/202	11:27	124
	Ananya	Kate	1	23/10/202	12:15:00	130
	Jim	Kate	0	23/10/202	12:20	231
	Abhigya	Kate	1	23/10/202	12:26	205
	Abhigya	Kate	1	23/10/202	12:30	395
	Jill	Mike	2	23/10/202	12:39	260
	Mia	King	0	23/10/202	12:45	120
	Jim	King	1	23/10/202	12:54	276
	Abhigya	Mike	2	23/10/202	13:1	190
	Mia	Kate	1	23/10/202	13:35	155
	Mike	Mia	0	23/10/202	13:41	133
	Phoebe	Jill	0	23/10/202	13:48	140
	Phoebe	Kate	2	23/10/202	14:46	165

Final Leaderboard after **15** iterations:

	B	C	D	E	F	G
	Name	Score	Match no	Wins	hortest Tim	Win%
	Mike	2	2	2	190	100
	Ananya	2.5	3	2	130	66.66667
	Jim	1.5	2	1	276	50
	Abhigya	2.5	5	2	205	40
	Mia	2	3	1	155	33.33333
	Jill	1.5	3	1	124	33.33333
	Kate	1.5	6	1	165	16.66667
	Phoebe	0.5	3	0	999	0
	King	0.5	2	0	999	0
	Mike	0.5	1	0	999	0

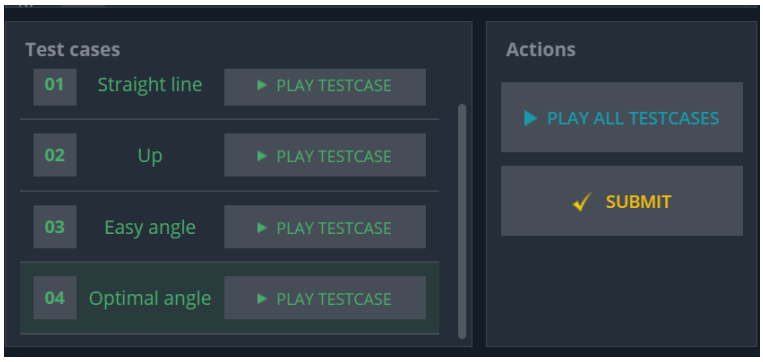
CODING GAMES ASSIGNMENTS

Game-1:

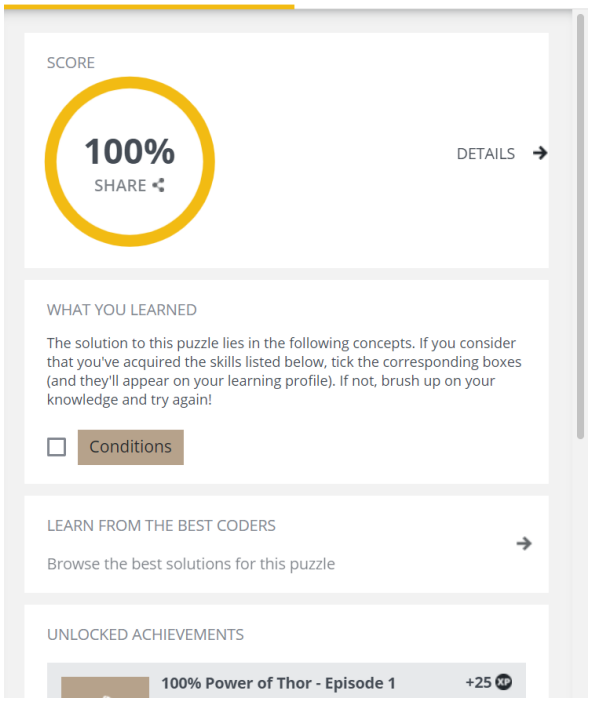
```
import sys
import math

# Auto-generated code below aims at helping you parse
# the standard input according to the problem statement.
# ---
# Hint: You can use the debug stream to print initialTX and initialTY, if Thor seems not follow you
r orders.

# light_x: the X position of the light of power
# light_y: the Y position of the light of power
# initial_tx: Thor's starting X position
# initial_ty: Thor's starting Y position
light_x, light_y, initial_tx, initial_ty = [int(i) for i in input().split()]
thor_x, thor_y = initial_tx, initial_ty
# game loop
while True:
    remaining_turns = int(input())  # The remaining amount of turns Thor can move. Do not remove th
is line.
    direction_x = ""
    direction_y = ""

    if thor_x == light_x:
        if thor_y > light_y:
            direction_y = "N"
            thor_y -= 1
        elif thor_y < light_y:
            direction_x = "S"
            thor_y += 1

    elif thor_y == light_y:
        if thor_x > light_x:
            direction_x = "W"
            thor_x -= 1
        elif thor_x < light_x:
            direction_y = "E"
            thor_x += 1

    elif thor_x != light_x and thor_y != light_y:
        if thor_y > light_y and thor_x > light_x :
            direction_y = "NW"
            thor_x -= 1
            thor_y -= 1
        elif thor_y > light_y and thor_x < light_x :
            direction_y = "NE"
            thor_x += 1
            thor_y -= 1
        elif thor_y < light_y and thor_x > light_x :
            direction_x = "SW"
            thor_x -= 1
            thor_y += 1
        elif thor_y < light_y and thor_x < light_x :
            direction_x = "SE"
            thor_x += 1
            thor_y += 1

    # Write an action using print
    # To debug: print("Debug messages...", file=sys.stderr)
```

```
    # A single line providing the move to be made: N NE E SE S SW W or NW
    print(direction_y + direction_x)
```

(Code Snippet - Any Style)



("Run Code" Artifact on Codin games Platform)



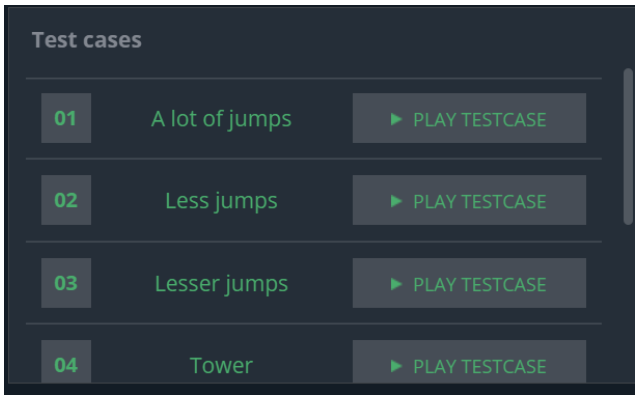
Game-:2

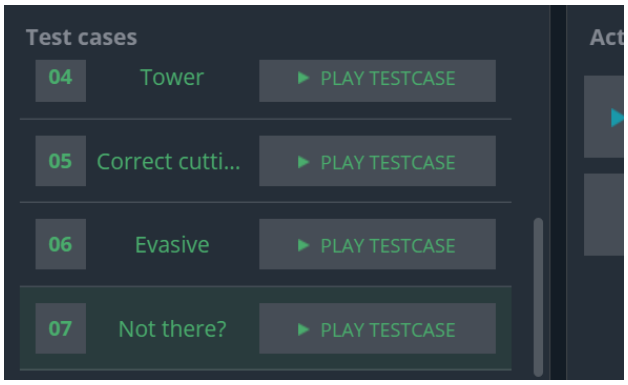
```
import sys

w, h = [int(i) for i in input().split()]
n = int(input())  # maximum number of turns before game over.
x, y = [int(i) for i in input().split()]
first_x = 0
last_x = w
first_y = 0
last_y = h

# game loop
while True:
    # the direction of the bombs from batman's current location:
    # U, UR, R, DR, D, DL, L or UL
    direction = input()
    if 'U' in direction:
        last_y = y - 1
    elif 'D' in direction:
        first_y = y + 1
    if 'L' in direction:
```

```
        last_x = x - 1
    elif 'R' in direction:
        first_x = x + 1
    x = (first_x + last_x) // 2
    y = (first_y + last_y) // 2
    print("{} {}".format(x, y))
```

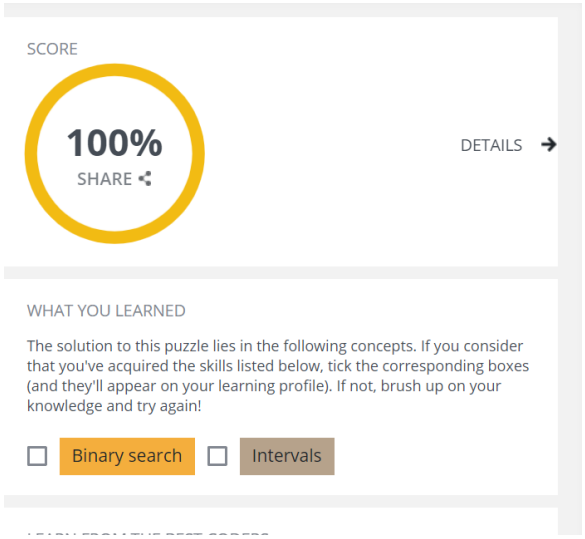
(Code Snippet - Any Style)

Solved test cases





Submitted



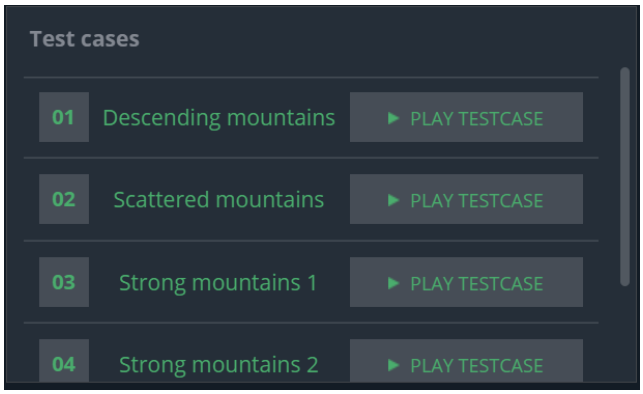
Game-:3

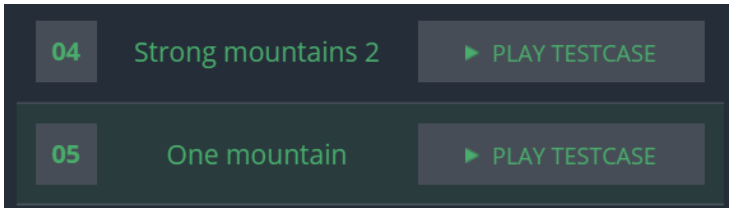
```
import sys

#Game loop.
while True:
    max = 0
    maxIndex = -1

    for i in range(8):
        #Read inputs.
```

```
        mountainH = int(input())

        #Set highest mountain.
        if mountainH > max:
            max = mountainH
            maxIndex = i

    #Output highest mountain.
    print(maxIndex)
```

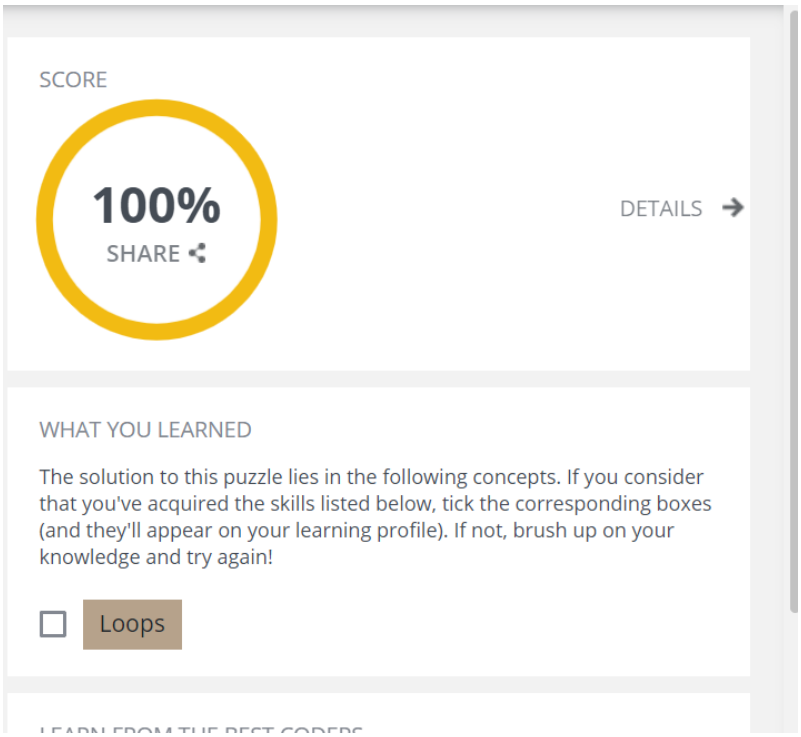
(Code Snippet - Any Style)

Solved test cases





Submitted



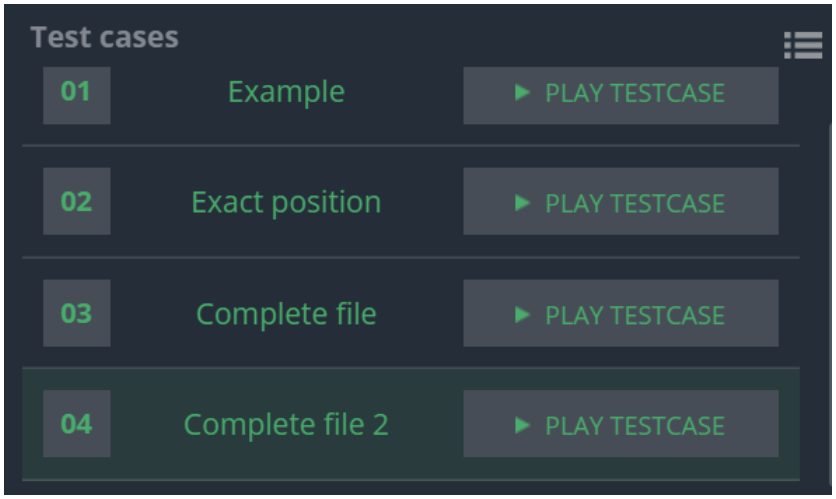
Game-:4

```python
import sys
import math

#Calculates the distance between two points on earth.
def distance(lonA, latA, lonB, latB):
    lonA = math.radians(lonA)
    lonB = math.radians(lonB)
    latA = math.radians(latA)
    latB = math.radians(latB)

    x = (lonB - lonA) * math.cos((latA + latB) / 2)
    y = latB - latA

    return math.sqrt(x * x + y * y) * 6371

#Read inputs.
lonA = float(input().replace(',', '.'))
latA = float(input().replace(',', '.'))
N = int(input())

#Define minimum.
min = sys.maxsize
minName = ''

for i in range(N):
    #Read defibrillator.
    DEFIB = input().split(';')

    lonB = float(DEFIB[4].replace(',', '.'))
    latB = float(DEFIB[5].replace(',', '.'))

    #Calculating distance for current defibrillator.
    d = distance(lonA, latA, lonB, latB)

    #Set nearest defibrillator.
    if d < min:
        min = d
        minName = DEFIB[1]
```

```
#Print nearest defibrillator.
print(minName);
```

(Code Snippet - Any Style)

Solved test cases



Submitted