import random  
  
*# Function to create the minefield*def create\_minefield(size, num\_mines):  
 field = [[" " for \_ in range(size)] for \_ in range(size)]  
  
 *# Place mines randomly* mines = set()  
 while len(mines) < num\_mines:  
 x, y = random.randint(0, size-1), random.randint(0, size-1)  
 mines.add((x, y))  
  
 for (x, y) in mines:  
 field[x][y] = "\*"  
  
 return field, mines  
  
*# Function to count adjacent mines*def count\_adjacent\_mines(field, x, y):  
 size = len(field)  
 count = 0  
 for i in range(x-1, x+2):  
 for j in range(y-1, y+2):  
 if 0 <= i < size and 0 <= j < size and field[i][j] == "\*":  
 count += 1  
 return count  
  
*# Function to print field for the player*def print\_field(field, revealed):  
 size = len(field)  
 for i in range(size):  
 row = ""  
 for j in range(size):  
 if revealed[i][j]:  
 if field[i][j] == "\*":  
 row += " \* "  
 else:  
 row += f" {field[i][j]} "  
 else:  
 row += " # "  
 print(row)  
 print()  
  
*# Game logic*def play\_minesweeper(size=5, num\_mines=5):  
 field, mines = create\_minefield(size, num\_mines)  
 revealed = [[False for \_ in range(size)] for \_ in range(size)]  
  
 while True:  
 print\_field(field, revealed)  
 x = int(input(f"Enter row (0-{size-1}): "))  
 y = int(input(f"Enter col (0-{size-1}): "))  
  
 if (x, y) in mines:  
 print("💥 BOOM! You hit a mine. Game Over!")  
 break  
  
 revealed[x][y] = True  
 field[x][y] = count\_adjacent\_mines(field, x, y)  
  
 *# Check win condition* if all(revealed[i][j] or field[i][j] == "\*" for i in range(size) for j in range(size)):  
 print("🎉 Congratulations! You cleared the minefield!")  
 break  
  
*# Run the game*play\_minesweeper()