CSPP Week - 4 Exam

Section - II | Time: 3 hours | Max Score: 25 pts

Bingo Game Programming

Overview
In this assignment, you will be developing a simple text-based version of Bingo using Python. The game will include board creation, board display, number marking, and player interaction. You'll write five functions to accomplish these tasks.

Function 1: generateBoard() - 3 pts
Write a function called generateBoard() that doesn't take any arguments and returns a 2D list representing the Bingo board. Each cell of the board should contain a unique random integer between 1 and 100.

Hint: You can use random.sample(range(1, 101), 25) to generate 25 unique random numbers between 1 and 100.

```
generateBoard()
# Output
[[23, 11, 42, 9, 37],
 [12, 56, 89, 34, 90],
 [64, 27, 18, 77, 55],
 [49, 44, 51, 26, 10],
 [76, 70, 2, 91, 31]]
```

Function 2: displayBoard(board) - 3 pts
Write a function called displayBoard() that takes a 2D list board as its argument and prints the board in a readable format. Replace any marked numbers with 'X'. The function should return None.

```
displayBoard([[23, 11, 42, 9, 37],
              [12, 'X', 89, 34, 90],
              [64, 27, 18, 77, 55],
              [49, 44, 51, 26, 10],
              [76, 70, 2, 91, 31]])
```

The output should look like below on the console

```
23 11 42 9 37
12 × 89 34 90
64 27 18 77 55
49 44 51 26 10
76 70 2 91 31
```

Function 3: markNumber(board, number) - 3 pts
Write a function called markNumber() that takes a 2D list board and an integer number as its arguments. The function should replace the number on the board with 'X' if it exists. The function should return None.

Function 4: getUserNumber() - 4 pts
Write a function called getUserNumber() that prompts the user for a number that has been drawn in Bingo. Make sure to validate that the input is an integer between 1 and 100. Keep prompting until you get a valid input.

```
Enter the drawn number: test
Invalid input! Please enter an integer between 1 and 100.
Enter the drawn number: 150
Invalid input! Please enter an integer between 1 and 100.
Enter the drawn number: 25
# Returns 25
```

Function 5: checkWin(board) - 4 points
Write a function checkWin(board) that takes a board (a 2D list) and returns a Boolean. The function should return True if there is a row, column, or diagonal completely filled with 'X'; otherwise, return False.

Winning Scenario
In Bingo, the goal is to have numbers called out that match those on your board in a specific arrangement. For this simplified text-based version of the game, let's focus on a single win condition:

A player wins when any one of the rows, columns, or the two diagonals is completely marked off with 'X'.

Function 6: playBingoGame() - 8 pts
Write a function called playBingoGame() that ties everything together.

Steps:

a. **Setup:** Welcome the player and generate a new board. Display the starting board to the user using generateBoard() and displayBoard().

b. **Game Loop:** Create a while-loop that continues until someone achieves Bingo. You can write a separate function to check for Bingo if needed.

c. **User Input:** In each loop iteration, prompt the user to enter the number drawn using getUserNumber().

d. **Mark and Display:** Use markNumber() to mark the drawn number on the player's board. Then, display the updated board.

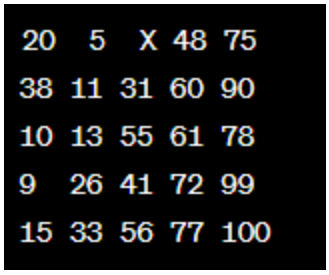
e. **Winning:** Use checkWin(board) to find if a player achieves Bingo, print out a congratulatory message and tell them how many rounds it took to win.

f. **Turn Counter:** Keep track of the number of rounds it took for the player to win and display this when the game ends.

Example Game Walkthrough
Assuming your board starts like this (generated randomly):

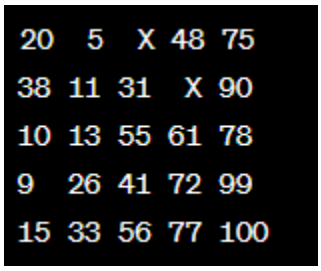
```
Board at Start:
20  5 17 48 75
38 11 31 60 90
10 13 55 61 78
9  26 41 72 99
15 33 56 77 100
```

Round 1: The user is prompted to enter a drawn number. They enter "17".
The function markNumber() replaces "17" with "X":

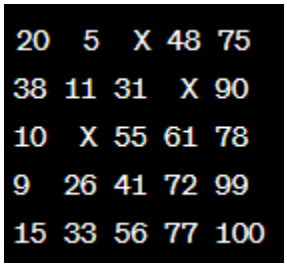


Round 2: User enters "60".

The function markNumber() replaces "60" with "X":

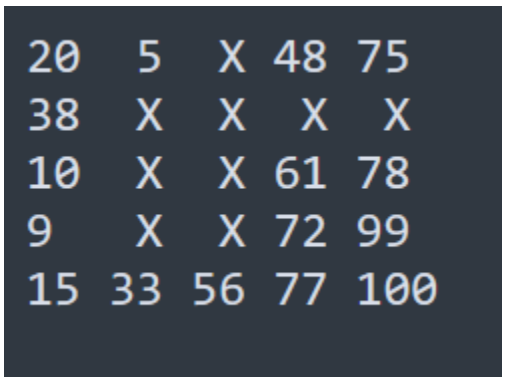


Round 3: User enters "13".
The function markNumber() replaces "13" with "X":

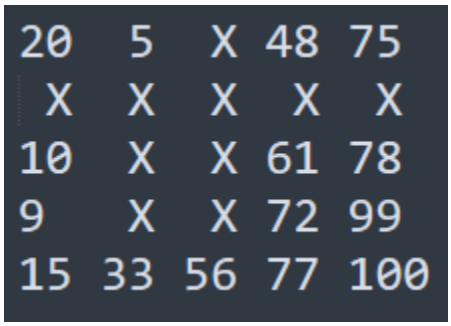


…..

Round N: The board eventually looks like this:



When the user enters "38", markNumber() replaces it with an "X":



At this point, the function checking for a win detects that the second row is entirely filled with "X".
Winning: The game prints a congratulatory message and informs the player that they have won after N rounds.