**Cinema Ticket Preorder App**

**Name:** Bryan Yadiel Caban Rodriguez

**Date Created:** Jan 13, 2025

**Program Description:** A program that pre-sells a limited number of cinema tickets. Each buyer can buy up to 4 tickets, and no more than 20 tickets can be sold. It prompts the user for the desired number of tickets and t hen displays the number of remaining tickets after their purchase. Repeats until all tickets have been sold. Then, displays the total number of buyers.

**Functions used in the Program:**

1. **Function Name:** runcode()

**Description:** Checks if the user wants to run the program by prompting for a yes/no response.

**Parameters:** None

**Variables:**

* runcode (str) - stores the user's input on whether to run the program

**Logical Steps:**

1. Prompt the user with a question asking if they want to run the code
2. Convert the user's input to lowercase using casefold()
3. Return the processed input

**Returns:** str - The user's response in lowercase ('y' or 'n')

1. **Function Name:** get\_tickets\_purchased()

**Description:** Prompts the user for the number of tickets they want to purchase and validates the input to ensure it's between 1 and 4.

**Parameters:** None

**Variables:**

* tickets (int) - stores the number of tickets the user wants to purchase

**Logical Steps:**

1. Create an infinite loop to keep asking until valid input is received
2. Try to convert user input to an integer
3. Validate if the number is between 1 and 4
4. Return valid input or display error message for invalid input
5. Catch ValueError for non-numeric input

**Returns:** int - The number of tickets requested (1-4)

1. **Function Name:** sell\_tickets(remaining\_tickets, total\_buyers)

**Description:** Processes the ticket sale by updating the remaining tickets and total buyers count.

**Parameters:**

* remaining\_tickets (int) - Current number of available tickets
* total\_buyers (int) - Current count of people who have purchased tickets

**Variables:**

* tickets\_purchased (int) - Number of tickets the current buyer wants to purchase

**Logical Steps:**

1. Call get\_tickets\_purchased() to get number of tickets requested
2. Check if enough tickets are remaining
3. If sufficient tickets available:
   * Subtract purchased tickets from remaining tickets
   * Increment total buyers
   * Display purchase confirmation and remaining tickets
4. If insufficient tickets:
   * Display error message

**Returns:** tuple (int, int) - Updated remaining\_tickets and total\_buyers

1. **Function Name:** main()

**Description:** Main function that manages the ticket selling process until all tickets are sold.

**Parameters:** None

**Variables:**

* total\_tickets (int) - Total number of tickets available (20)
* remaining\_tickets (int) - Number of tickets still available
* total\_buyers (int) - Number of people who have purchased tickets

**Logical Steps:**

1. Initialize variables for total tickets (20), remaining tickets, and total buyers
2. Create a loop that continues while tickets remain available
3. Call sell\_tickets() to process each sale
4. Update remaining tickets and total buyers
5. Display final message when all tickets are sold

**Returns:** None

**Overall Program Logical Flow:**

1. Program starts and asks if user wants to run the code (runcode())
2. If yes, main() is called:
   * Initializes ticket tracking variables
   * Enters selling loop until all tickets are sold
   * Each sale involves:
     + Getting number of tickets wanted (get\_tickets\_purchased())
     + Processing the sale (sell\_tickets())
     + Updating tracking variables
   * Displays final results
3. If no, displays "Code did not run" message

**Link to your repository:** <https://github.com/xXTeinsXx/COP2373>