ASIA PACIFIC UNIVERSITY ESPORTS CHAMPIONSHIP MANAGEMENT SYSTEM

(Working Duration: Monday of Week 10 to Monday of Week 14 – 30 Marks)

STORY BACKGROUND:

The Asia Pacific University Esports Championship (APUEC) has become one of the most anticipated collegiate esports tournaments in Asia, attracting top gamers from universities across the region. Hosted annually by Asia Pacific University (APU), this event is more than just a gaming competition, it celebrates skill, teamwork, and strategic thinking.

Given the scale and complexity of the tournament, efficient management is essential. APU has taken an innovative approach by leveraging advanced data structures to enhance scheduling, player management, spectator experience, and match tracking. The tournament committee has identified key areas where **Stacks, Queues, Priority Queues, and Circular Queues** can optimize operations for a seamless competition.

CHALLENGES:

Managing an esports championship requires handling multiple components efficiently, including:

- **Match Scheduling & Player Progression,** ensuring players compete fairly through qualifier, group, and knockout stages.
- **Tournament Registration & Player Queueing,** managing sign-ups, check-ins, and last-minute entries efficiently.
- **Live Stream & Spectator Queue Management,** prioritizing VIPs, streamers, and audience engagement.
- **Game Result Logging & Performance History,** keeping an accurate record of match outcomes, stats, and highlights.

SYSTEM TASKS & MEMBER ALLOCATION:

Each student in a team will take responsibility for one of the following tasks, applying appropriate data structures to solve key challenges.

TASK 1: MATCH SCHEDULING & PLAYER PROGRESSION

This task involves organizing the competition's structure and managing player advancement through stages such as qualifiers, group rounds, and knockout matches.

Key Actions:

- Ensure fair match pairings based on player rankings.
- Manage the flow of matches.
- Update tournament brackets dynamically as players progress.

TASK 2: TOURNAMENT REGISTRATION & PLAYER QUEUEING

Handling player registrations, last-minute check-ins, and ensuring an orderly tournament start.

Key Actions:

- Implement check-ins for players.
- Prioritize early-bird signups and wildcard entries.
- Handle withdrawals and replacements efficiently.

TASK 3: LIVE STREAM & SPECTATOR QUEUE MANAGEMENT

Ensuring a well-managed viewing experience for streamers, VIP guests, and general spectators.

Key Actions:

- Implement seatings for VIPs and influencers.
- Manage audience overflow efficiently to avoid congestion.
- Organize dedicated viewing slots for live-streaming setups.

TASK 4: GAME RESULT LOGGING & PERFORMANCE HISTORY

Recording match outcomes, player statistics, and keeping a historical record for future reference.

Key Actions:

- Store recent match results for quick review.
- Maintain a structured history for player performance tracking.
- Provide easy access to past results for tournament analysis.

Lab Work #2 – Program & Live Presentation Guidelines (30 Marks)

Team & Submission Requirements

1. Each team must have up to **FOUR (4)** members.

2. Your team is required to use **C++** to develop **ONLY ONE (1)** prototype in this section, without using STL containers like `<list>` or `<vector>`.

3. Each team member must be responsible for **AT LEAST ONE (1)** of the following tasks, applying the appropriate data structures and algorithms.

   - **Task 1: Match Scheduling & Player Progression**
   - **Task 2: Tournament Registration & Player Queuing**
   - **Task 3: Live Stream & Spectator Queue Management**
   - **Task 4: Game Result Logging & Performance History**

   Marks awarded will be based on individual contributions, considering each member's responsibility in the system and how accurately you can justify your selection of data structures and algorithms.

4. The evaluation criteria for this lab work #2 also include assessing the clarity and structural design of the code, as well as the quality of comments and adherence to good programming practices. (e.g., indentation, meaningful identifier names, comments, etc.).

5. **This task requires a group submission, but grading will be based on everyone's contribution to the system.**
   The team leader must upload a ZIP file of the system solution to the Moodle system by **Monday of Week 14**, no later than 5:00 pm.

   - Include only `.cpp`, `.hpp`, and `.csv`/text files.
   - Follow this file naming format: `<GroupNo>_<TeamLeaderID>_<1stMemberID>_<2ndMemberID>_<3rdMemberID>.zip`
   - For example, "`G1_TP012345_TP014556_TP067554_TP034325.zip`"

   Refer to **Page 6** for marking criteria of this Lab Evaluation Work #2 submission.

6. After submitting your system code to Moodle, your team must schedule ***a live presentation*** with your lecturer *between Tuesday of Week 14 to Friday of Week 16 (Your final exam week #1)*.

Summary: What You Need to Submit?

1. This assignment requires **TWO (2)** submissions by your team, which include the following:

   i. Group Submission (By Monday, Week 14, before 5:00pm)
   - C++ solution in a ZIP folder, inclusive *the .cpp, .hpp and csv/text files*.

   ii. Individual Live Demonstration (30 Marks)
   - Each member must present their task **individually** within **30 minutes**, including both **system demonstration and Q&A session**.
   - PowerPoint slides are not required for this demonstration.
   - Failing to attend the **live demo = 0 marks** for Lab Work #2.

MARKING CRITERIA

(Lab Evaluation Work #2 - 30 MARKS)

This Lab Evaluation Work #2 will be assessed based on the following **INDIVIDUAL** performance criteria:

Assessment Components	Inclusive	30 Marks
CLO3: Lab Evaluation Work #2 – Individual Development Skills		
Practical Skills: Problem-Solving Skills (15 Marks)		
• Identify and address technical challenges.	Assessment of Problem-Solving Ability	
• Use of data structures and algorithms.	Technical Proficiency	
• Implementation of features according to design specifications.	Technical Proficiency	
• Code quality, including readability, efficiency, and correctness.	Code Quality Evaluation	
• Quality of individual contributions relative to team goals.	Contribution Assessment	
• Innovation and creativity in developing and implementing features.	Creativity and Innovation Evaluation	
Practical Skills: Q&A with Justification of Data Structures (15 Marks)		
• Clear and logical explanation for the choice of data structures.	Justification Ability	
• Relevance of chosen data structures to the functionality implemented.	Relevance Evaluation	
• Justification aligned with the system requirements and performance needs.	Alignment with Requirements	
• Effectiveness of the live presentation in explaining individual contributions	Presentation Effectiveness	