

**Operating System Lab**

**CSE 412**

**Project Report on**

**Quiz Game Using Operating System Concepts**

**Submitted To:**

**Tahmina Islam**

Lecturer, Department of CSE

University of Information Technology & Sciences

**Saima Siddique Tashfia**

Lecturer, Department of CSE

University of Information Technology & Sciences

#### **Submitted By:**

1. Israt Jahan Eva (7B1\_57)

2. Irin Sultana Poly (7B2\_91)

3. Sadia Akter (7B1\_59\_)

4. Asadur Rahman Asif (7B2\_79)

**Date of Submission:** 10/12/24

**Introduction:**

The "Quiz Game Using Operating System Concepts" project is an interactive, console-based program written in C. This project demonstrates the use of operating system concepts such as process handling, randomization, and memory management. The quiz game is designed to simulate question handling, dynamic user interaction, and feedback, showcasing how core principles of operating systems can be applied to real-world problem-solving.

The game enables users to engage with randomly selected questions while ensuring no repetition. It highlights how operating system techniques such as random number generation, loop control, and memory tracking can create a seamless and dynamic user experience. This project also provides valuable insights into practical programming and operating system utilities in application development.

**Objectives:**

The objectives of this project are as follows:

1. **Operating System Integration:** To implement a program that applies concepts like randomization and memory allocation for question selection and management.
2. **Dynamic Question Handling:** To simulate a system that selects questions randomly without repetition, mimicking real-time process scheduling.
3. **User Feedback Mechanism:** To provide real-time feedback and performance tracking based on user inputs.
4. **Error Handling and Robustness:** To ensure the program can handle invalid inputs gracefully and continue execution without crashing.
5. **Practical Learning:** To demonstrate practical use of arrays, loops, and conditional statements within an operating systems context.

**Methodology:**

The methodology for the "Quiz Game Using Operating System Concepts" project can be summarized as follows:

#### **1. Initialization of Questions and Options:**

* Define an array of questions, each with four answer options.
* Maintain a parallel array for correct answer indices.
* Use structured data to organize questions and answers for scalability.

#### **2. Random Question Selection:**

* Use the rand () function to generate random indices, simulating process scheduling.
* Ensure no question is repeated during the game session by maintaining a history of selected indices.
* Implement a loop to validate uniqueness of each selected question.

#### **3. User Interaction:**

* Display a question and its options, prompting the user to select an answer.
* Include error-checking mechanisms to handle invalid user inputs (e.g., non-numeric inputs or out-of-range choices).
* Validate the user’s input against the correct answer index and provide immediate feedback.

#### **4. Feedback and Scoring:**

* Dynamically update the user’s score for correct answers.
* Provide personalized feedback for each question, indicating whether the answer was correct or incorrect.
* Track the user’s performance over the session to display cumulative results.

#### **5. Final Summary:**

* Display the total score and feedback based on the user’s performance.
* Include motivational messages based on the score range (e.g., excellent, good, or needs improvement).
* Ensure the program exits cleanly, simulating proper process termination.

**Implementation:**

#### **Code Overview:**

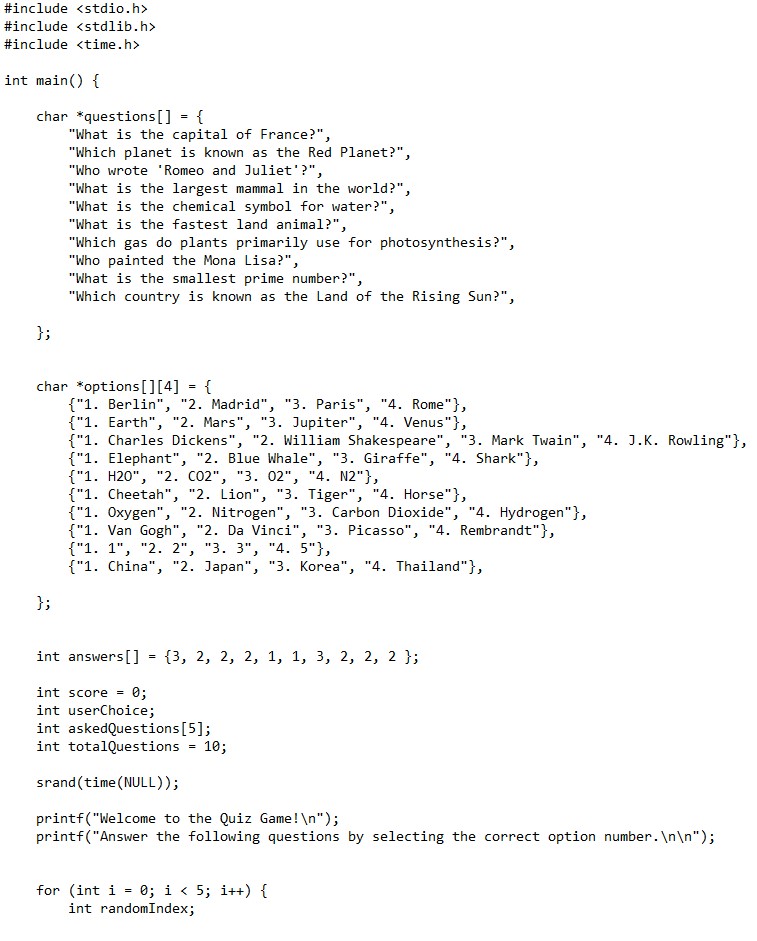
The project code is structured as follows:

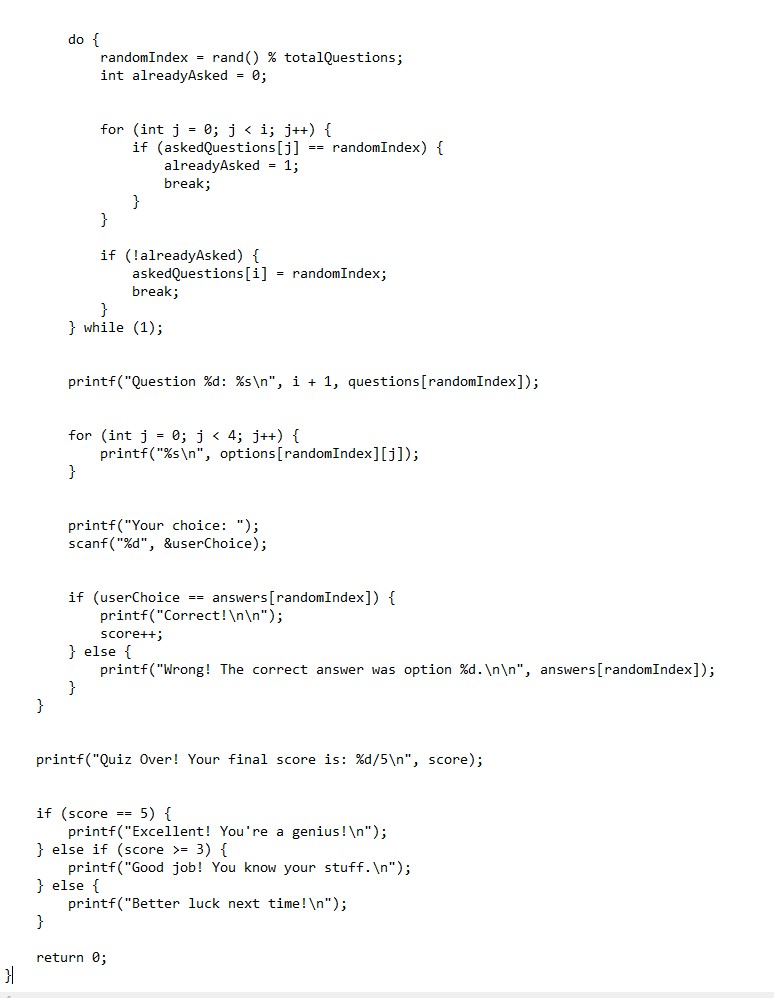
1. Define arrays for questions, answer options, and correct answers.
2. Utilize randomization to select unique questions, simulating scheduling.
3. Implement input handling for user responses, including validation for robustness.
4. Validate answers and update scores dynamically.
5. Summarize results at the end of the session, providing user feedback and motivational commentary.

#### **Features:**

* **Randomization:** Uses seeded random numbers for question selection to ensure variability across sessions.
* **Dynamic Feedback:** Provides real-time feedback for each answer, making the game interactive.
* **Error Handling:** Handles invalid inputs gracefully without terminating the program unexpectedly.
* **Performance Tracking:** Tracks the user’s cumulative score and displays detailed performance analysis.

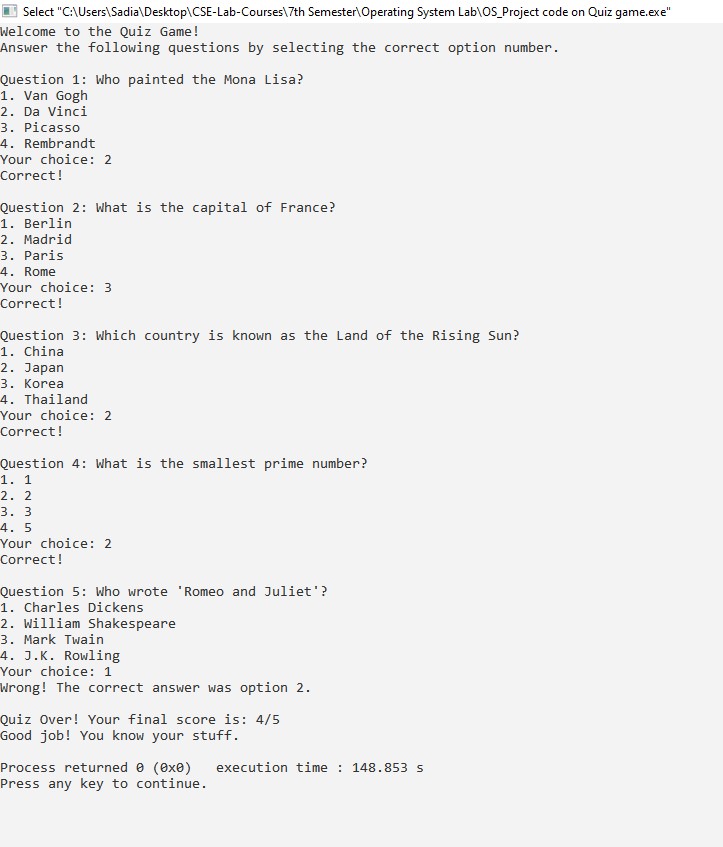
**Code Screenshot:**

****

****

**Output:**

The program provides the following outputs:

* A sequence of randomly selected questions with four answer options each.
* Real-time feedback on the correctness of each answer.
* User-friendly prompts to guide the gameplay.
* A summary of the user’s final score along with motivational feedback such as:
  + **Excellent:** For scores of 5/5.
  + **Good Job:** For scores of 3 or 4.
  + **Better Luck Next Time:** For scores below 3.

**Result:**

The "Quiz Game Using Operating System Concepts" successfully integrates operating system principles into an interactive application. By simulating random question selection and feedback mechanisms, the project highlights the practical application of process management and memory utilization. The game provides a seamless and engaging user experience, demonstrating the effective use of OS concepts in a real-world scenario.

Additionally, the program's ability to handle user errors and provide dynamic interaction ensures robustness, making it a practical tool for learning and testing fundamental concepts of operating systems and C programming.

**Conclusion:**

In conclusion, the "Quiz Game Using Operating System Concepts" project effectively applies fundamental operating system principles to create a dynamic and interactive application. This project serves as an excellent example of how theoretical concepts can be utilized to build practical solutions. By integrating randomization, memory management, and feedback mechanisms, the game showcases the versatility and relevance of operating systems in programming.

The project not only enhances understanding of process handling and error management but also provides a foundation for building more complex systems using OS principles. The interactive and user-friendly nature of the program makes it a valuable educational tool.

The End