I. Report Structure (PDF)

1. **Title Page**
    - Project Title
    - Team Members' Names
    - Course Information
    - Submission Date
2. **Table of Contents**
    - List all sections and subsections with page numbers.
3. **Introduction**
    - Briefly introduce the project and its objectives.
    - State the significance of the project.
4. **Problem Explanation**
    - Include flowcharts or diagrams in your final report to represent the system's architecture and logic. Additionally, for each module of your C# console application, include the following:

    4.1. **Module Description**
    - Purpose and function of the module.
    - Explanation of the problem it addresses.

    4.2. **Algorithm Analysis**
    - Step-by-step description of the algorithm used.
    - Explanation of the logic and flow of the algorithm.

    4.3. **Code Explanation (Pseudocode)**
    - Provide pseudocode for the module to illustrate its structure and logic.
5. **Design Choices & Results**

    5.1. **Design Choices**
    - Discuss the design decisions made throughout the project.
    - Justify the choice of algorithms, data structures, and methodologies used.

    5.2. **Results**
    - Present the outcomes and effectiveness of your solutions.
    - Use summary or comparative tables for performance metrics.

- Discuss the implications of the results and any observed patterns.

6. **Task Distribution**
   - Include a table clearly showing who did what in your team, including responsibilities such as report writing, implementation, testing, and design.
   - Ensure all team members participate in the various aspects of the project, as point deductions will apply if this is not clearly indicated.
   - Include a table that specifies the contributions of each team member, such as:
     - Report Writing
     - Implementation
     - Testing
     - Design
7. **Conclusion**
   - Summarize the key findings of the project.
   - Discuss potential future work or improvements.
8. **References**
   - List all sources and references used in the report.
9. **Appendices**

II. Formatting Guidelines

- Font: Times New Roman, size 12.
- Length: Maximum of 15 pages.
- Tables: Ensure all results and comparisons are presented in well-organized tables.

III. C# Console Application Code: Your C# console application code should be included in the compressed file along with your report. The project package must adhere to the following:

1. **Code Folder:**
   - Include all relevant C# code files organized in meaningful folders.
   - Provide any supporting files or resources needed for your application to run smoothly.

2. **Code Execution:**
    - Ensure that the C# console application runs without errors. The instructor will not debug or correct your code, so it's your responsibility to make sure the program functions properly.

Notes:

1- Good Coding Practices

Follow good software development practices, including proper use of comments, indentation, and modularity in your code.

2- **Guidance on Visualization:** Given the limitations of console applications in representing visual data, consider the following suggestions for effective visualizations:

- **Text-based Visualization:** Use simple text-based diagrams to illustrate the flow of data and processes within your application. For example, you can visually represent the steps taken from question input to answer extraction using arrows or lists.
- **Progress Indicators:** Implement simple progress indicators or status updates within the console to inform users of the current step in the processing pipeline (e.g., "Analyzing question...", "Retrieving relevant sentences...", "Extracting answer...").
- **Formatted Output:** Use formatted console output (e.g., using different colors for questions and answers) to create a visually distinct presentation of information. This can make it easier for users to follow along with the process.