1. Read a .txt resume file and print its full content.
2. Read a .txt file and display only the first 300 characters as a preview.
3. Given extracted text, convert it to lowercase and remove extra spaces.
4. Given a file path, detect whether the file is PDF, DOCX, or TXT.
5. Load two text files (resume and job description) and confirm successful loading.
6. Extract text from a PDF resume using a Python PDF library.
7. Extract all paragraphs from a DOCX resume and combine them into a single string.
8. Create a unified function load_file(path) that reads PDF, DOCX, and TXT files.
9. Using the unified loader, display the resume preview (first 500 characters) and job description preview (first 500 characters).
10. Clean the extracted text after parsing and display both raw and cleaned versions.
11. Simulate file upload by accepting resume and job description file paths and parsing both.
12. Handle error cases such as unsupported file types, empty files, and invalid file paths.
13. Store parsed resume and job description text in a structured dictionary.
14. Save cleaned resume and job description text into a JSON file.
15. Parse multiple resumes of different formats and store their cleaned text in a list.
16. Build a universal parser that detects file type, extracts text, cleans text, and displays a preview.
17. Parse multiple resumes along with one job description and generate previews for each resume.
18. Create a separate module that handles only parsed document preview formatting.
19. Replicate the complete Milestone-1 output including file upload simulation, parsed document preview, and cleaned text generation.
20. Create an end-to-end script that performs file loading, parsing, cleaning, preview display, and saving parsed output.