Honors/Bonus Project: Markdown Compiler

Background

In this project, you will implement a Markdown language compiler that uses HTML as a target language. The project is worth 100 points, which will be calculated using the rubric below. To receive the Honor's credit, you should score at least 70 points out of 100. The students who use this project to receive the Honor's credit cannot use it to receive bonus points. The students who do not use this project for receiving the Honor's credit can use it for receiving 10–100 bonus points.

Project Description

You are expected to implement a compiler for the GitHub flavor of Markdown language. The target language is properly-formatted HTML. You should implement the project using RPLY lexer and parser. Specifically, you are expected to use Python to develop a list of lexemes and associated patterns, write the BNF grammar of the language, and implement production functions for the BNF grammar. The use of intermediate representation is optional. *Unlike other class projects, you are expected to conduct your own research related to technical details of the project.* The instructor and TA will help you clarify the requirements of the project, but you should implement and troubleshoot the project by yourself, without anyone's help.

Implementation

Your implementation should include a single Python 3.6+ file called **`md2html.py`**. The program takes a single command-line argument — the Markdown input file, which must have the **`.md`** extension. The output of the program should be the file with the same base name but the extension changed into **`.html`**. For example, if the input file is **`README.md`**, then the compiler will be used as follows:

```
python3 md2html.py README.md
```

After running the program, the output is expected to be saved in the **`README.html`** file. Invoking **`md2html.py`** without an argument or using a file extension other than **`.md`** should trigger a proper error message.

IMPORTANT: The failure to fully follow the above implementation guidelines or avoiding using RPLY for compilation will result in 0 (zero) points scored for the project.

Grading Rubric

The project will be graded based on the following rubric. Bonus point students cannot submit re-grading requests for this assignment. The Honor's students can submit re-grading requests, but not later than

December 6, 2023. Unlike other projects in the class, you cannot resubmit the project: once it is submitted, it will immediately be considered as ready for grading. The last day you can submit the project is December 11, 2023. The following is the grading rubric for the project.

Requirement	Points
The compiler produces a properly-formatted HTML code, which includes the **`<!DOCTYPE ..>`**, **`<html>`**, **`<body>`**, and **`<p>`** elements	10
The compiler properly implements 6 levels of headings compiled to <h1> … <h6> elements	10
The compiler property implements bold, italic, and strikethrough text	10
The compiler properly implements the nested italic and bold-italic elements	10
The compiler implements the quoted text elements using formatting of your choice	10
The compiler implements the in-sentence (single-tick) code and code blocks (triple-tick) using formatting of your choice involving a cross-platform monospace font	10
The compiler properly implements hyperlinks	10
The compiler properly implements images, both local and remote	10
The compiler properly implements bulleted lists	10
The compiler properly implements numbered lists	10

Please submit the solution to D2L.