Functional Requirements

The interpreter should behave like a simple command-line interface (CLI) that can read commands from standard input and write results to standard output.

Lexical Rules:

- Commands, options, and arguments are separated by whitespace characters (' ' and '\t').
- An argument enclosed in quotation marks (") is treated as a single unit, regardless of spaces inside.
- Special commands for input/output redirection ('<', '>', '>>') and pipelines ('|') have special meaning, unless they are inside quotation marks.

Input and Output Management for Commands:

- A command may receive input from:

• Standard input (stdin)
• A quoted argument
• A file
• Input redirection ('<')

- A command may send output to:

• Standard output (stdout)
• Output redirection ('>' or '>>')

Supported Commands:

- echo: Prints given arguments to standard output.
- prompt: Changes the shell prompt string.
- time: Prints the current time.
- date: Prints the current date.
- touch: Creates an empty file or updates an existing file’s timestamp.
- truncate: Empties the contents of an existing file.
- rm: Deletes a specified file.
- wc: Counts the number of lines, words, and characters in input.
- tr: Translates or removes characters in input.
- head: Prints the first N lines of input (default: 10).
- batch: Executes commands from a file sequentially.

Pipelines:

- Multiple commands can be combined with the pipeline operator ('|').
- The output of one command becomes the input of the next command.

Error Handling:

- Lexical, syntactic, and semantic errors must be reported with appropriate error messages.

- When an error occurs, the interpreter should print the message and stop executing the current line.