Validation_patterns

End of work
\nBest Practices for Implementing Input Validation Securely:\n- Validate All Inputs: Treat all data originating from external sources (user input, APIs, external files) as untrusted.\n- Validate at the Trust Boundary: Perform validation as early as possible, ideally when data enters your system.\n- Use Positive Validation (Whitelisting): Define and enforce a list of allowed characters, patterns, or values rather than trying to block malicious input (blacklisting).\n- Enforce Minimum and Maximum Lengths: Define and enforce minimum and maximum length constraints for all inputs to prevent buffer overflows and other attacks.\n- sanitize Input: Remove or escape any special characters or sequences that could be interpreted as commands or code by the application or underlying systems.\n- Use Built-in Validation Functions: Whenever possible, use built-in validation functions or libraries provided by the programming language or framework, as they are often more robust and secure.