[Code] Execution	**[Code Execution Issues]** - The code does not compile or throws runtime errors	**[Code Runtime Side Effects]** - The code runs, but has side-effects such as warnings, print statements that were used for debugging, etc	- Code runs perfectly without any errors or warnings
[Code] Output	**[Incorrect Code Output]** - Output is irrelevant or incorrect - Output is incomplete	N/A	- Output perfectly aligns with the requirements of the prompt
[Code] Performance	**[Major Code Performance Issues]** - The code implementation is extremely inefficient. Ex: takes an O(n^3) brute force approach when it's possible to fulfill the request in O(nlogn)	**[Minor Code Performance Issues]** - The code implementation is moderately efficient with room for further optimization. Ex: O(n^2) was used when O(nlogn) is possible	- The code implementation is well-optimized, utilizing efficient algorithms and data structures wherever possible
[Code] Readability	- **[Major Code Readability Issues]** The code is difficult to read because of poor formatting such as missing indentation, poor markdown, excessive white space, or no whitespace (minified code), etc; OR - **[Misleading Code Variable Names]** Variable/class/method names are not indicative of their function. Ex: a misleading variable name such as `even_array = [1, 3, 5, 7]`	- **[Minor Code Readability Issues]** The code can be formatted better in some areas, but it's still readable OR - **[Poor Code Variable Names]** Variable/class/method names don't follow the general naming conventions of the respective language	- The code is well organized and uses consistent formatting, making it highly readable AND - Variable/class/method names are meaningfully chosen and are reflective of their purpose