|  |  |  |
| --- | --- | --- |
| Peer Review Form | | |
| Student Name |  | |
| Reviewer Name |  | |
| Submitted Filename |  | |
| Evaluation Criteria | Check Point | 드롭다운 Result |
| Adherence to Common Constraints | All code within main() function? | Pass |
| Source filename correct? | Pass |
| Only standard output functions used? | Pass |
| ANSI C coding style followed? (specific examples if no) | Pass |
| Compilation and Execution | Compiles without errors or warnings? (attach error log if no) | Pass |
| Executable created successfully? | Pass |
| Program terminates gracefully? | Pass |
| Compiler Version Check | gcc 9.4.x or higher detected? (provide version info if yes) | Pass |
| Output Format Compliance | Output matches specified format exactly? (specific discrepancies) | Pass |
| Overall Assessment | Rate the code quality: (Excellent / Good / Average / Needs Improvement) | Excellent |
| Provide specific feedback on:   * Code Readability: (e.g., variable names, comments, structure) * Code Efficiency: (e.g., algorithms used, potential optimizations) * Functionality: (e.g., correctness, completeness) |  |
| Please provide detailed feedback, including areas for improvement and specific aspects you appreciated.  Example Feedback: "The code is well-structured and easy to follow. Good use of comments to explain the logic. However, the output format for the 'Introduction' section doesn't quite match the specification. (Specific example)."  "The bonus binary representation is a clever touch! However, there's a small error in the binary value for (specific number). (Details)." |  |

Important Notes:

* Be objective and fair in your assessment.
* Focus on providing actionable and constructive feedback.
* Highlight both strengths and areas for improvement.

This peer review form is designed to help students evaluate each other's code and learn from the process.