

# 브레인 스토밍 결과

## Project Brainstorming Result

Project Name	코드 난독화 도구 제작
-----------------	--------------

15 조

202002562 조인우

202002508 손지웅

201902686 노형우

지도교수: 조은선 교수님 (서명)

# Document Revision History

---

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2025/03/28	아이디어 수렴 작성	조인우
2	2025/03/28	시각화하기	손지웅
3	2025/03/28	아이디어 발산, AI 도구 활용 정보	노형우

# Contents

---

1.	(문제 해결을 위한) 아이디어 발산	4
2.	아이디어 수렴	4
3.	시각화하기	5
4.	AI 도구 활용 정보	5

## 1. (문제 해결을 위한) 아이디어 발산

코드 난독화	<ul style="list-style-type: none"> <li>소스 코드</li> <li>바이너리</li> </ul>	역공학	LLVM	<ul style="list-style-type: none"> <li>ANTLR</li> <li>Listener</li> </ul>
보안, 정보보호	miniC	컴파일러 <ul style="list-style-type: none"> <li>parser</li> <li>AST</li> </ul>	더미코드	치환
무작위	흐름 변형	인코딩 <ul style="list-style-type: none"> <li>Base64</li> </ul>	디버깅 도구 무력화	부작용

## 2. 아이디어 수렴

핵심 개념	정의 및 하위 개념 목록
코드 난독화	<ol style="list-style-type: none"> <li>소스 코드 난독화</li> <li>바이너리 난독화</li> </ol>
난독화 기법	<ol style="list-style-type: none"> <li>배치 난독화</li> <li>자료 난독화</li> <li>제어 흐름 난독화</li> <li>방지 난독화</li> </ol>
난독화 목적	<ol style="list-style-type: none"> <li>지적 재산, 핵심 알고리즘 보호</li> <li>라이선스 및 인증 우회 방지</li> <li>모바일 앱 보안, 웹 프론트엔드, 클라이언트 보호</li> <li>역공학 방지</li> </ol>
구현 방식(개념)	<ol style="list-style-type: none"> <li>프레임워크, 프로그래밍 언어</li> <li>컴파일러 구조, 컴파일러 개념</li> <li>제어 흐름 그래프</li> </ol>
부작용	<ol style="list-style-type: none"> <li>성능저하</li> <li>유지보수 난이도 상승</li> <li>가독성 저하</li> <li>각종 도구와 프레임워크 호환 불가 가능성</li> <li>바이너리 용량 증가</li> </ol>

