

TEAM WORK DIVISION — PLAGIARISM CHECKER PROJECT (PDF DOCUMENT)

PROJECT MODULE SPLIT (3-MEMBER TEAM)

1. YOU (Lead Coder) — Module C: Algorithms & Integration

Responsibilities:

- Tokenization logic
 - N-gram generator
 - N-gram similarity
 - Code preprocessing (remove comments, whitespace normalization)
 - Frequency vector creation for code
 - Cosine similarity
 - Integrating all modules into main.c
-

2. MEMBER 1 — Module A: Preprocessing & File Handling

Responsibilities:

- Read .txt and .c files
 - Lowercase conversion
 - Remove punctuation
 - Remove extra spaces
 - Return cleaned text buffers
-

3. MEMBER 2 — Module B: User Interface & Reporting

Responsibilities:

- Main menu creation
- Report file formatting
- Writing similarity scores, filenames, timestamps
- README.md documentation
- Tests and examples

FOLDER STRUCTURE

```
project/
  └── src/
      ├── moduleA_preprocessing.c (Member 1)
      ├── moduleB_report_and_menu.c (Member 2)
      ├── moduleC_algorithms.c (You)
      └── main.c (You)

  └── include/
      ├── preprocessing.h
      ├── report.h
      └── algorithms.h

  └── input/
  └── output/
  └── README.md
```

WHY THIS STRUCTURE WORKS

- Each member works independently.
 - Very low chance of merge conflicts.
 - Beginners get meaningful tasks.
 - You focus on complex algorithmic logic.
 - Fast integration before deadline.
-

SUMMARY

This division keeps the project balanced, clear, and doable within 10 days.