# RISC-V Code Generation Challenge Report

## Directory Structure

codegen-challenge/  
├── generate\_header.py # Python script to parse YAML and emit C header  
├── input.yaml # Original YAML data (from RISC-V Unified Database)  
├── data\_gen.h # Auto-generated C header with #defines  
├── main.c # C program to re-emit YAML using header constants  
├── output.yaml # Final YAML output after C program runs

## Objectives Achieved

| Goal | Status |
| --- | --- |
| Parse structured YAML input | Achieved |
| Auto-generate C header with defines | Achieved |
| Write C program to emit YAML format | Achieved |
| Round-trip YAML → Header → YAML transformation | Achieved |
| Compile & run C code using GCC | Achieved |
| Output verified against original input | Achieved |

## Step-by-Step Instructions

### 1. Install Dependencies

sudo apt update  
sudo apt install gcc python3 python3-pip  
pip3 install pyyaml

### 2. Generate the C Header from YAML

python3 generate\_header.py input.yaml > data\_gen.h

**Result**: data\_gen.h with constants like:

#define PKG\_0\_NAME "markdown-it\_markdown-it"  
#define PKG\_0\_VERSION "13.0.1"  
...

### 3. Compile the C Program

gcc main.c -o emit\_yaml

**Result**: Creates executable emit\_yaml

### 4. Run the Program

./emit\_yaml

**Output:**

output.yaml generated successfully.

### 5. Verify Final Output

cat output.yaml

**Output:**

- package: markdown-it\_markdown-it  
 version: 13.0.1  
 commit: e843acc9edad115cbf8cf85e676443f01658be08  
 date: May 3, 2022  
 notes:  
 - Rename variables that use python built-in names...  
 - Convert JS `for` loops to `while` loops...  
 ...

## Key Learnings

* Cross-language code generation (YAML → Python → C).
* Real-world YAML parsing with nested fields.
* C file I/O and macro management.
* Simulation of compiler-style codegen workflows.
* Debugging missing macro errors in C builds.