**WEEK 10**

**KU ID: 100058537**

**NAME: Mariam Alzaabi**

|  |
| --- |
| Q1.  Please write your GitHub Repository link for your week 10 Rustlings submission: |
| ANSWER:  https://github.com/mariamsz38/COSC320 |
| Q2.   1. Locate/find **1 significant struct in C2RUST?** (Important: Please also **write the URL**s where you find that **struct**) 2. explain what is the purpose of that **struct you have found from number (a)?**. 3. Show which modules/functions are dependent on this struct (from number (a)) and how they interact with it? |
| ANSWER:  **a. Significant Struct and URL:** One significant struct in C2RUST is Translation from the c2rust-transpile crate. URL: <https://github.com/immunant/c2rust/blob/master/c2rust-transpile/src/translator.rs> (Look for the pub struct Translation definition in this file.)  **b. Purpose of the Struct:** The Translation struct is a core component of the C2RUST transpiler. Its purpose is to maintain the state of the translation process as C code is converted into Rust. It keeps track of:   * All items generated so far (like functions, types, and variables translated from C to Rust), * The language features used during translation, * Associations between C declarations and their corresponding Rust identifiers, * Configuration settings for the translation process. Essentially, it acts as a central "context" or "environment" that the transpiler uses to manage and coordinate the conversion of an entire C program into Rust, ensuring consistency and correctness across the translation.   **Modules/Functions Dependent on This Struct and Interaction:**   * **Module:** translator module (in c2rust-transpile/src/translator.rs)   + **Function:** Translation::new()     - **Interaction:** This function initializes a new Translation instance with a given configuration. It's the starting point for any translation process.   + **Function:** Translation::translate()     - **Interaction:** This method uses the Translation struct to perform the actual translation of C AST (Abstract Syntax Tree) nodes into Rust code. It updates the struct's state as it processes declarations and statements. * **Module:** c\_ast module (in c2rust-transpile/src/c\_ast.rs)   + **Interaction:** The Translation struct consumes C AST data produced by this module. For example, it maps C declarations to Rust items, relying on the parsed C AST stored within its state. * **Module:** rust\_ast module (in c2rust-transpile/src/rust\_ast.rs)   + **Interaction:** This module provides helper functions that the Translation struct uses to generate Rust code. For instance, Translation calls these helpers to construct Rust AST nodes, storing the results in its internal collections. * **Example Interaction:** When translating a C function, Translation::translate() takes a C AST node, uses c\_ast to understand its structure, updates its internal state (e.g., adding a new Rust function item), and then uses rust\_ast helpers to generate the final Rust code. The struct's fields (like items for storing translated items) are modified throughout this process. |