**WEEK 10**

**KU ID: 100059726**

**NAME: Obaid Saif Alhmoudi**

|  |
| --- |
| Q1.  Please write your GitHub Repository link for your week 10 Rustlings submission: |
| ANSWER: |
| 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:   1. c2rust-transpile/src/convert\_type.rs ( https://github.com/immunant/c2rust/blob/3e0183e4d56f9d4d003f1ea2e9a814648debf307/c2rust-transpile/src/convert\_type.rs)   pub struct TypeConverter {  pub translate\_valist: bool,  renamer: Renamer<CDeclId>,  fields: HashMap<CDeclId, Renamer<FieldKey>>,  suffix\_names: HashMap<(CDeclId, &'static str), String>,  features: HashSet<&'static str>,  }   1. The purpose of the struct is to translate C types into Rust types during transpilation. This is done by managing type conversions from C to rust, ensuring correct naming by handling name collisions and renaming types as needed. Also maintaining field mapping for structs and unions to avoid duplicate names.   This struct is important because every C type is passed through Typeconverter to generate valid, safe and correct rust code. Without TypeConverter, the transpiler would not be able to resolve types so this ensures naming consistency. Also managing complex tupe mapping between C and rust.   1. In c2rust-transpile/src/translator/mod.rs This is one module I found that uses TypeCOnverter. But almost all modules have to use it     Sow now Translation owns a TypeConverter and uses it throughout its function to translate C types into Rust types  In the function translate:    Here are some interactions with TypeConverter.   * declare\_decl\_name() is called to register names for Rust type generation * alias\_decl\_name() is called to avoid name conflicts when multiple names refer to one type |