**Coding challenge**

A Rust program which contains a “Pairs” structure with **generic member types** named “old” and “new”, capable of storing any type of data primitive. These members represent current and previous values used for a configuration setting in an application.

A single, standardised method needs to be implemented for the structure called “**has\_changed**” (callable on any initialised instance of the structure) which returns a Boolean, irrespective of the data types involved.

If the values are the same, false is returned. If the values have indeed changed over time, true is returned.

**Design notes**

* No user input is required, use pairs of matched-typed literals to test.
* Test for Strings, integers, and float pairings.
* Note. You may need to constrain the generic type as only **partial equivalent checks** are possible (i.e. equality may not be achievable with all data types).

**Extension Tasks**

* Create a test module in Rust and check your function’s accuracy.
* Use **Cargo Test** to run the selected test module and confirm results.