Programming tasks

* Please create a public github repo
* Email me at ‘philip.rice2@ing.com` the details of the repo
* Put any comments you want to make in the README or in the email

Task 1 (estimated time << ½ hour)

* Rewrite this to use awk instead of sed

find $dir -name config.json | sed **'s#\/config.json##g'**

**find $dir -name config.json |awk '{gsub("/config.json", "", $0); print}'**

* + explain what this is doing
    1. This script will find the file (config.json) in the directory
    2. The output of find command will be **./config.json**. This will be piped to sed command
    3. sed command will replace /config.json to blank for all occurrences, as g is used.
    4. Output will be **.**
  + explain how you could write a test for this logic
    1. Use a test framework like Bats to create and assert test cases
    2. Use basic AAA pattern for testing -> Arrange, Act and Assert
    3. Create a test file test.sh and call the above command (ideally it will be in a sh file)
    4. Assert the equality for the expected output (.) to the actual value

Task 2 (estimated time << ½ hour)

* Rewrite this to use ten threads

**find -name config.json | xargs -P10 -n1 ./transform.sh {}**

* find $sourceDir -name pom.xml -exec transform.sh {}  \;
  + explain what this is doing

The script will find the file pom.xml in the directory and will do the operations on the file defined in transform.sh

* + explain how you could write a test for this logic

1) As the script is performing operations on a file , the test case will depend on the logic in transform.sh

2) Create a test case file test.sh and also a sample file test\_pom.xml.

The sample file will be created on the fly and deleted after the test case execution.

3) Run the test case on the sample file. Then assert the operations performed on the file

with the expected output.

* 1. Delete the sample test\_pom.xml

Task 3 (estimated time 1 to 4 hours depending on familiarity and access to environment)

* This is the bulk of the time. It is actually just a wrapper around curl, but requires access to azure devops. It is very straightforwards to create your own organization and project inside azure and costs no money. HCL may already have an environment. If that’s too awkward you can do something similar with github.

* Create a bash script called ‘ali’ that does the following
  + If called with -help it describes what it does
  + If called with incorrect parameters it follows normal scripting conventions
  + If called with the parameters `ali git get repoId branchId filePath’ it loads the file from Azure Devops
    - See [https://docs.microsoft.com/en-us/rest/api/azure/devops/git/items/get?view=azure-devops-rest-5.1](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Frest%2Fapi%2Fazure%2Fdevops%2Fgit%2Fitems%2Fget%3Fview%3Dazure-devops-rest-5.1&data=02%7C01%7Cgaurav.mohan%40hcl.com%7C107873ba3197487180d908d7aee4ffa3%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C637170170868337484&sdata=9kwC9xpYdYRDYiVspL900lNzjLmUXC%2BSabAAX68OvhY%3D&reserved=0)
  + If called with ‘ali git get repoId branchId filePath --debug’  it will not actually go to azure devops but will instead print to the ‘standard out’ the curl command it uses
  + Consider how the apikey and username should be passed to the script (make a choice, and use that choice)

The api key and username can be securely keep in Azure Key vault. It will need a separate shell script to interact with Azure subscription and get the access to api keys, certificates, username, passwords etc

* If it is going to take too long and you are uncomfortable with the task, please do it with pseudo code