Important Instructions

1. The codebase needs to be shared via github.com
2. Third party python libraries are allowed only for -
   - An http client
   - Ascii/plaintext table library
   - Python web framework (like flask etc.)
   - Python library to connect the mock server via SSH. *DO NOT* use a library to parse the ouput.
   - For web app frontend in *Task 4*, any CSS/JS framework is fine.
3. The test will be assessed on code quality parameters like design, project layout, unittests etc.

Task 1

Setup mock router (SSH)

- Setup cisshgo (https://github.com/tbotnz/cisshgo) locally.
- Replace the `show_running-config.txt` file with the one shared with this task.

Task 2

Webservice/API

1. Build a webservice in python
2. It should connect the cisshgo service (in Task 1) via SSH in real time
3. Run `show running-config` command
4. Parse it's output

5. Return all the details in this (JSON) format.

```
[{'interface': interface_name,
  'ip_address': ip_address,
  'status': status,
  etc. ...
 },
 ...]
```

Example:

Block:

```
interface GigabitEthernet0/0
 description to-LAN-2
 ip address 172.16.2.1 255.255.255.0
!
```

Output:

```
[{'interface': 'GigabitEthernet0/0',
  'ip_address': '172.16.2.1',
  'subnet': '255.255.255.0',
  'description': 'to-LAN-2'
 }]
```

NOTE: Text blocks in the input are separated by "!" (exclamation marks).

You will then need to serve this data over a rest API.
The API will have these endpoints:

- An endpoint to return all `interface` blocks
- Another endpoint which will return the data only for one `interface`. The `interface` name will be supplied in the URL itself (not as a querystring parameter).

Task 3

An API client

Write a python program that will test the above webservice by making requests to both the end points and print the info in a plaintext(ascii) table to `stdout`.

Task 4

Requirement story:
As a user, I should be able to input an `interface` name and then view the corresponding interface details in the browser.

Build a **web app** for this story. Use the APIs built in Task 2.