You are required to implement a python functions that:

- Generate membership/non-membership proofs
- Verify a generated membership/non-membership proofs

Download starter code to begin with. In the project you should find 5 python files:

- **1_merkle_leaves_gen.py:** this script is responsible for generation of different IDs that will be used as merkle tree leaves. You **won't change or run** this file.
- **2_merkle_root_calc.py:** this script is responsible for calculating merkle tree root and keep it so that it can be used in verification process of membership/non-membership. You **won't change or run** this file.
- **utils.py:** this script helpers functions for calculating hashes and dealing with merkle tree. You **won't change** this file.
- **3_membership_non_membership_proof_gen.py:** this script is responsible for generating membership/non-membership of a given value according to its presence/absence in the generated ids from `**1_merkle_leaves_gen.py**` file. If the value exists in generated id, a proof of membership is supposed to be generated. otherwise, a proof of non-membership is generated. You are **required to** implement the gen_membership_proof() and gen_non_membership_proof() functions. After finishing implementation.

  Run the script:

```
python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value 9 --proof_file membership_proof.json

python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value 10 --proof_file non_membership_proof.json

python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value -1 --proof_file min_proof.json

python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value 999999999999999999999999999999 --proof_file
max_proof.json
```

  After running the thes commands you should find 4 json files were created named 'membership_proof.json`, `non_membership_proof.json`, `min_proof.json` and `max_proof.json`. To verify your implementation you can compare your results with 9.json file and 10.json file attached in the project.

  Also, you are required to run the script for the value (your id % 99):

```
python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value {your id%99} --proof_file my_id_proof.json
```

  For example if your id is 190152384 (190152384 % 99 = 15) you will run script as below:

```
python 3_membership_non_membership_proof_gen.py --merkle_leaves_file
merkle_leaves.json --value 15 --proof_file my_id_proof.json
```

- **4_membership_non_membership_proof_verifier.py:** this script is responsible for verification of membership/non-membership proofs generated from **`3_membership_non_membership_proof_gen.py`** file. You are **required to** implement the verify_membership() and verify_non_membership() functions. After finishing implementation.
  Run the script:

```
python 4_membership_non_membership_proof_verifier.py --merkle_root_file
merkle_root.json --proof_file membership_proof.json

python 4_membership_non_membership_proof_verifier.py --merkle_root_file
merkle_root.json --proof_file non_membership_proof.json

python 4_membership_non_membership_proof_verifier.py --merkle_root_file
merkle_root.json --proof_file min_proof.json

python 4_membership_non_membership_proof_verifier.py --merkle_root_file
merkle_root.json --proof_file max_proof.json

python 4_membership_non_membership_proof_verifier.py --merkle_root_file
merkle_root.json --proof_file my_id_proof.json
```

  Running each command should display `Successfully verified!`.

You are required to submit your source code **including** membership_proof.json, non_membership_proof.json, min_proof.json, max_proof.json and my_id_proof.json.
Note: **Don't edit** the following files:

- merkle_leaves.json
- merkle_root.json