Create a new Python `virtualenv` and install `maturin` into the `virtualenv.`

```
$ mkdir secret_sharing
$ cd secret_sharing
$ python -m venv .env
$ cd .env
$ ./Scripts/activate
$ pip install maturin
```

Initialize a new RUST package with PyO3 bindings.

```
$ maturin init
```

Specify metadata about the new package in the manifest file “secret_sharing/ Cargo.toml”. Set the crate-type as “cdylib”, and add pyo3 to the dependencies.

```
[lib]
name = "secret_sharing"
crate-type = ["cdylib"]
[dependencies]
pyo3 = { version = "0.16.5", features = ["extension-module"] }
```

Annotate the RUST function with #[pyfunction] to turn it into a PyCFunction and wrap the result in a PyResult.

```
#[pyfunction]
fn reconstruct_private_key(filename: &str) -> PyResult<String> {
    let shares = 5;
    let threshold = 3;
    let private_key = reconstruct::<GF256, CompactShamir>(filename, threshold);
    Ok(private_key)
}
```

Add the function to the #[pymodule]

```
#[pymodule]
fn secret_sharing(_py: Python<'_>, m: &PyModule) -> PyResult<()> {
    m.add_function(wrap_pyfunction!(reconstruct_private_key, m)?)?;
    Ok(())
}
```

The above will expose the reconstruct_private_key function in the Python module called secret_sharing. Build the package and install it into the Python virtualenv previously created and activated.

```
$ maturin develop
```

The secret_sharing package is then ready to be used from Python.

```
$ python
>>> import secret_sharing
>>> secret_sharing.split_private_key("privateKey.txt")
Shares:
1|6ca345ec3b5f5432cf66e1e7dc8d1ff4ca7bad5ff254dcede49d31f60ffc1c7b
2|24c0c1fe5a27a8a0a8cf9fc4fba05e58b7521ea430bb44d2514850c6f535f675
3|86097316f7d1086c927700c7d1fe9afc52fafdba00abe3357edee7afca4dfe34
4|a7dbd65f52393310ee7ec03d8b0ce0475193b4ec6cc3c4e2ea36c80256e01b94
5|051264b7ffcf93dcd4c65f3ea15224e3b43b57f25cd36305c5a07f6b699813d5
>>> secret_sharing.reconstruct_private_key("shares.txt")
Private Key: ce6af70496a9f4fef5de7ee4f6d3db502fd34e41c2447b0acb0b869f3084143a
```