Homework 2

1. Try out the vanity bitcoin address example at asecurity or the Ethereum version

2. What do you understand by

    1. O(n) Computational time grows in proportion to input size n
    2. O(1) Computational time is constant regardless of input size
    3. O(log n) Computational time grows linearly while input size n grows exponentially

3. Which of those is best when describing a proof size

    O(1) for constant proof size

4. In your teams discuss why are there few identity projects on Starknet ?

For now ZK is used for scaling (roll-up) and not for privacy, L3s could be a good potential for ID projects

5. In preparation for future lessons you need to install Protostar

See documentation and this useful medium article

Linux / Mac

1.

```
curl -L https://raw.githubusercontent.com/software-
mansion/protostar/master/install.sh | bash
```

2. Restart the terminal.
3. Run `protostar -v` to check Protostar and cairo-lang version.
   It adds to your PATH in .bashrc, you may need to move that line to .bash_profile If it doesn't find protostar.
   You may also need to update your version of git

Windows

Windows is not supported.
You can either use a linux environment or virtual machine in Windows,
or use gitpod :

I have set up an environment on gitpod, please make sure this works for you
Go to http://gitpod.io

and use the repo https://github.com/ExtropyIO/CairoBootcamp

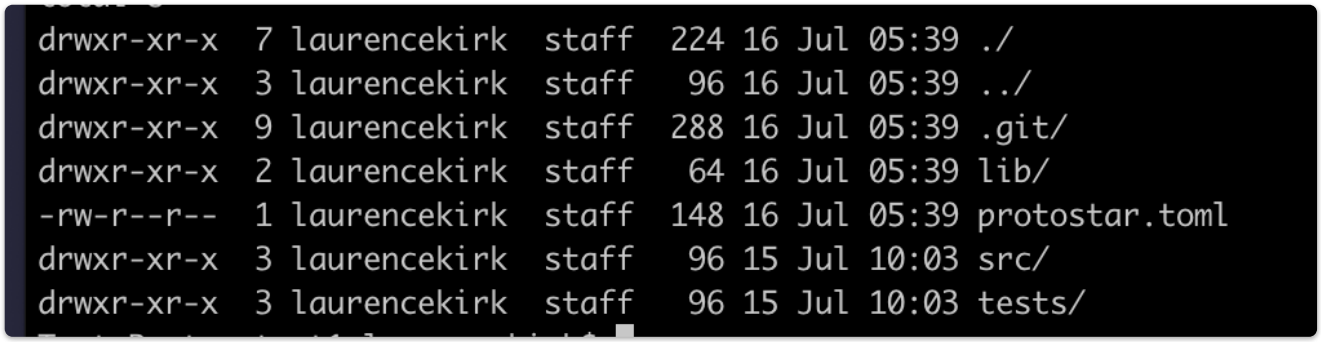
or go direct to URL
https://gitpod.io/#https://github.com/ExtropyIO/CairoBootcamp

---

To create a new project use

```
protostar init
```

This will give a directory structure similar to this



Configuration

This is specified in the .toml file

```
["protostar.config"]
protostar_version = "0.4.2"

["protostar.project"]
libs_path = "./lib"         # a path to the dependency directory

# This section is explained in the "Project compilation" guide.
["protostar.contracts"]
main = [
  "./src/main.cairo",
]
```

Compiling your programs / contracts

once you have specified the contracts in the protostar.toml file, run

```
protostar build
```

to compile them.

Deploying your programs / contracts

You need to specify the path to the compilation results.

```
$ protostar deploy ./build/main.json --network alpha-goerli
```

Testing your programs / contracts

Protostar will find the test file using its name, checking if it begins with `test_` prefix, and has `@external` functions, which names begin with `test_` .

A test looks like

```
@external
func test_sum{syscall_ptr : felt*, range_check_ptr}():
    let (r) = sum_func(4,3)
    assert r = 7
    return ()
end
```

You can run the tests, specifying the test directory

```
protostar test ./tests
```