Homework 13

You can use gitpod workspace
https://gitpod.io/#https://github.com/ExtropyIO/Academy
(https://gitpod.io/#https://github.com/ExtropyIO/Academy)

Navigate to the DeFiHardhat directory , there is the beginnings of a project
Run npm i to install the dependencies

In a terminal start ganache that forks the mainnet

```
npx ganache-cli -f <Your INFURA URL> --unlock
0x503828976D22510aad0201ac7EC88293211D23Da -p 8545
```

We are going to create a DeFi contract to interact with Uniswap V3 Router
Documentation for the router is available here
(https://docs.uniswap.org/protocol/reference/periphery/interfaces/ISwapRouter)

If you get stuck we are roughly following the example given here
(https://docs.uniswap.org/protocol/guides/swaps/single-swaps)

Before we call the swap function we need to

1. Import the router and transfer helper contracts from uniswap

```
import '@uniswap/v3-periphery/contracts/interfaces/ISwapRouter.sol';
import '@uniswap/v3-periphery/contracts/libraries/TransferHelper.sol';
```

add constants for the DAI and USDC addresses on the mainnet

```
address public constant DAI = 0x6B175474E89094C44Da98b954EedeAC495271d0F;
address public constant USDC = 0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48;
```

2. Set up a variable for the router interface ISwapRouter, call it swapRouter and in the constructor set the address as

```
0xE592427A0AEce92De3Edee1F18E0157C05861564
```

3. Write a function called swapDAItoUSDC to perform the swap it should have an input parameter of uint256 amountIn and an output parameter of uint256 amountOut.
4. In the swapDAItoUSDC function declare a variable called ***params*** of type

`ISwapRouter.ExactInputSingleParams`

to hold the parameters for the swap, and set these as follows :

```
tokenIn: DAI,
tokenOut: USDC,
fee: 3000,
recipient: msg.sender,
deadline: block.timestamp,
amountIn: amountIn,
amountOutMinimum: 0,
sqrtPriceLimitX96: 0
```

we need to approve the router to spend the contract's DAI, for this we can use the TransferHelper

```
TransferHelper.safeApprove(DAI, address(swapRouter), amountIn);
```

finally make the call to the router to perform the swap

```
amountOut = swapRouter.exactInputSingle(params);
```

Write unit tests for your code

Now we need to write tests for this

To run the tests run

```
npx hardhat test --network fork
```

1. Write a unit test in your DeFi.test.js file, this test should
   a) Send DAI to the DeFi contract, the amount it sends, and the input amount should be at least 99999999900000. Be aware of the number of decimal places in DAI and USDC.

   b) Check that your contract now has sufficient DAI for the swap
   c) Call your swapDAItoUSDC function to perform the swap
   d) Check the USDC balance of the owner account to see the result of the swap.

2. Add parameters to your function, so that you can specify the tokens you would like to exchange

3. Try exchanging DAI for other tokens such as UNI or AAVE