哈希的补充教学：

当我们需要计算哈希值时，Solidity 提供了常用的加密函数：

keccak256(bytes memory) returns (bytes32) 计算输入的 Keccak-256 散列。

sha256(bytes memory) returns (bytes32) 计算输入的 SHA-256 散列。

ripemd160(bytes memory) returns (bytes20) 计算输入的 RIPEMD-160 散列。

下面的例子就是计算“ABC“的 Keccak-256 散列值。当然你可以试试将 keccak256 换为 sha256 或 ripemd160。

```
pragma solidity ^0.5.0;

contract Test {
    function callKeccak256() public pure returns(bytes32 result){
        return keccak256("ABC");
    }
}
```

程序代码

```
pragma solidity ^0.5.0;

contract Test {      RETURN costs 0 gas - this line costs 123 gas - 90888 gas left
    function callKeccak256() public pure returns(bytes32 result){    276 gas
        return keccak256("ABC");
    }
}
```

部署情况

```
[vm] from: 0x5B3...eddC4 to: Test.(constructor) value: 0 wei data: 0x608...10032 logs: 0 hash: 0xd06...7aeb6
状态                    0x1 交易已打包且执行成功
交易哈希                0xd063c3ac274a742b5b27cebee9e5869dd877b2046c290dd2c4a20f49fbd7aeb6
区块哈希                0x8faf37f152c9f1d938701dd71acf43d8614bc58e3306c7f9555efc2c05a9cb03
区块号                  17
合约地址                0x5FD6eB55D12E759a21C09eF703fe0CBa1DC9d88D
from                    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                      Test.(constructor)
gas                     gas
交易成本                93505 gas
执行成本                37487 gas
输入                    0x608...10032
解码输入                []
解码输出                -
日志                    []
```

输出 ABC 的哈希

我们再加一个函数 judgment()，将输入的值与 callsha256 计算的"ABC"进行比较：

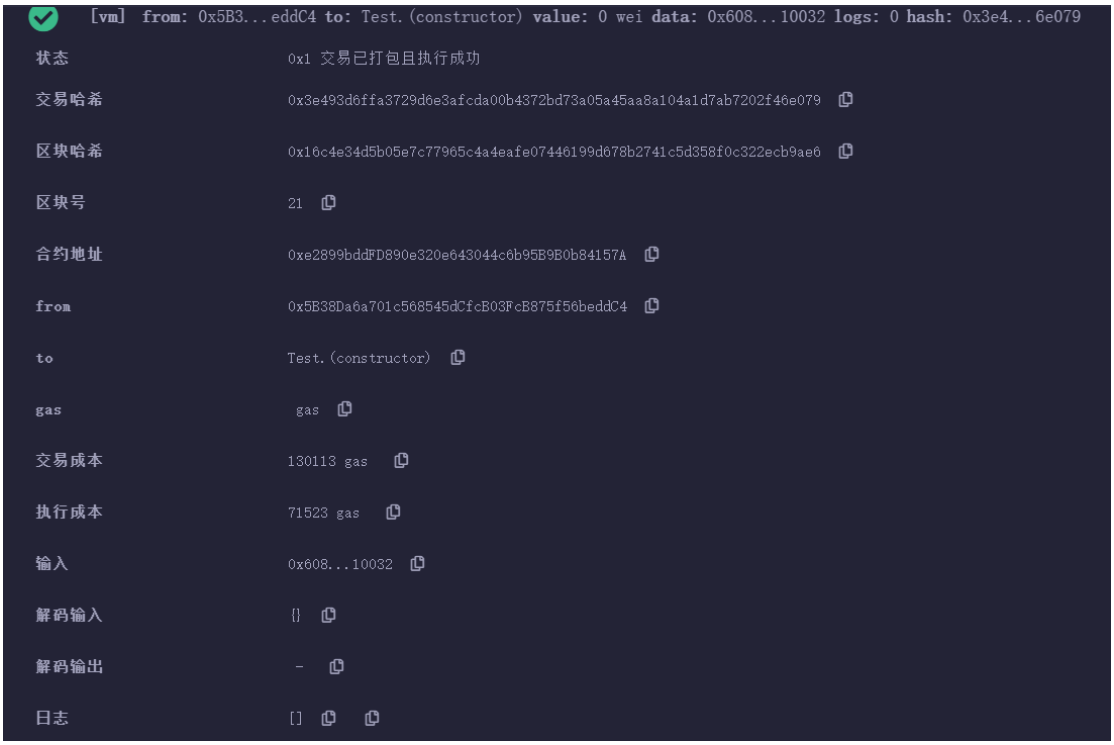
```
pragma solidity ^0.5.0;

contract Test {
    function callsha256() public pure returns(bytes32 result){
        return ripemd160("ABC");
    }

    function judgment(bytes32 trueOr) public returns(bool result){
        if(trueOr == callsha256()){
            return true;
        }
        else return false;
    }
}
```

思考一下，如何正确的填写 judgment 的输入参数呢？

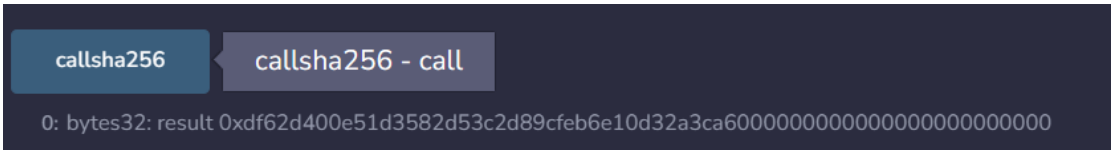
程序代码

```
pragma solidity ^0.5.0;

contract Test {      RETURN costs 0 gas - this line costs 123 gas - 90888 gas left
    function callsha256() public pure returns(bytes32 result){    infinite gas
        return ripemd160("ABC");
    }

    function judgment(bytes32 trueOr) pure  public returns(bool result){    infinite gas
        if(trueOr == callsha256()){
            return true;
        }
        else return false;
    }
}
```

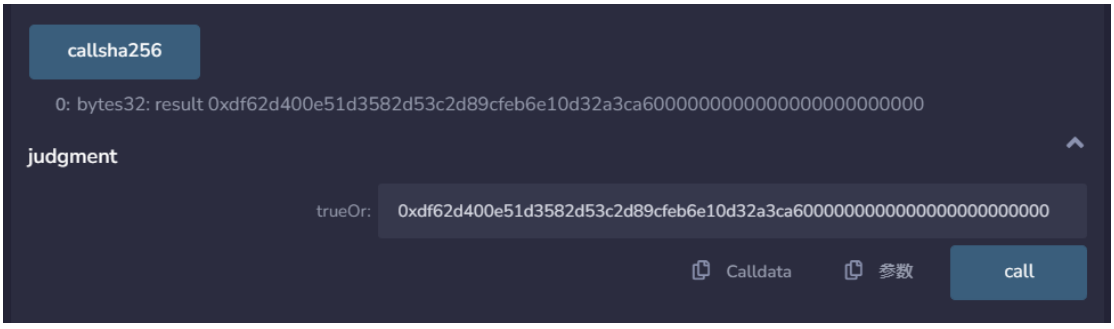
部署



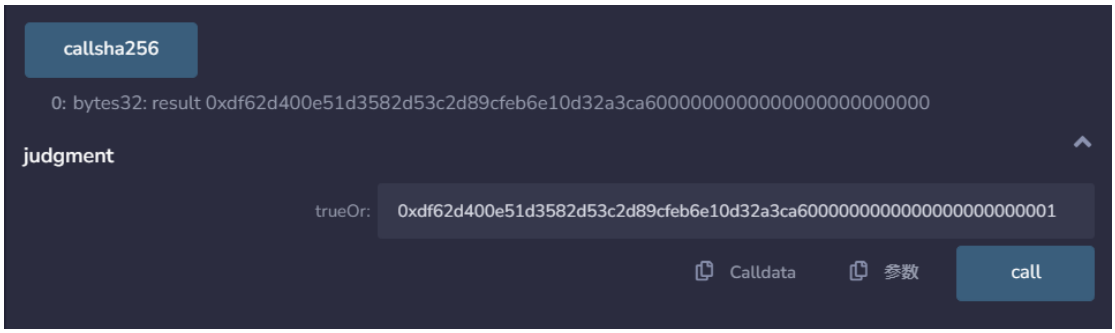
用 callsha256 返回结果



输入与这个结果相同的 16 进制到 judgment，返回结果为 true



输入不同的到 judgment，返回为 false



应该填入相应的 16 进制数