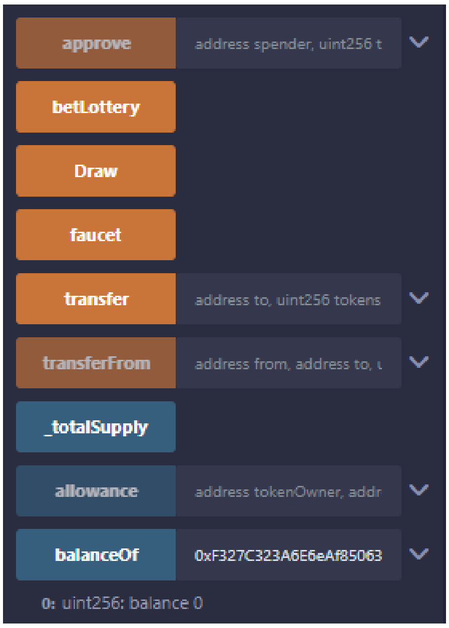
Introduction to Blockchain and Its Applications
Term Project 2 – smart contract

M11215032 葉品和

1. Create your own token and transfer 100 tokens to TA



address spender, uint256 t

address to, uint256 tokens

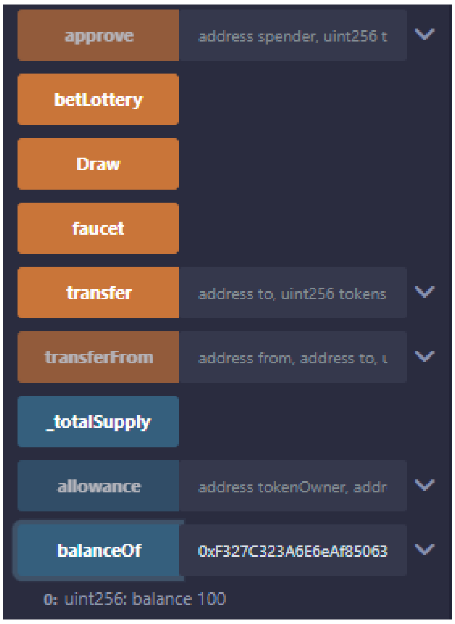
address from, address to, u

address tokenOwner, addr

balanceOf 0xF327C323A6E6eAf85063

0: uint256: balance 0

before transfer



address spender, uint256 t

address to, uint256 tokens

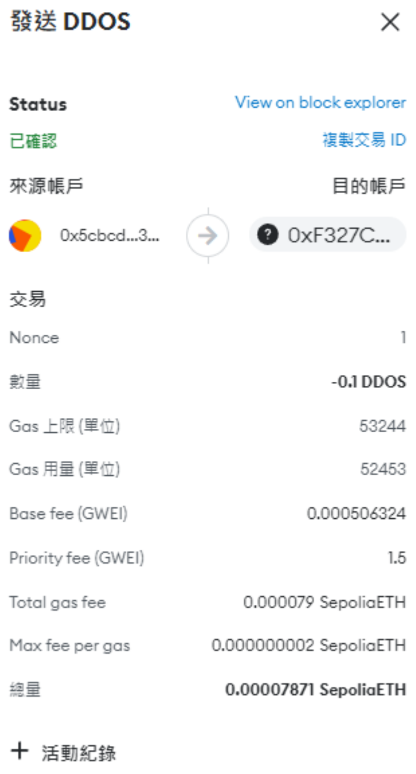
address from, address to, u

address tokenOwner, addr

balanceOf 0xF327C323A6E6eAf85063

0: uint256: balance 100

after transfer



發送 DDOS

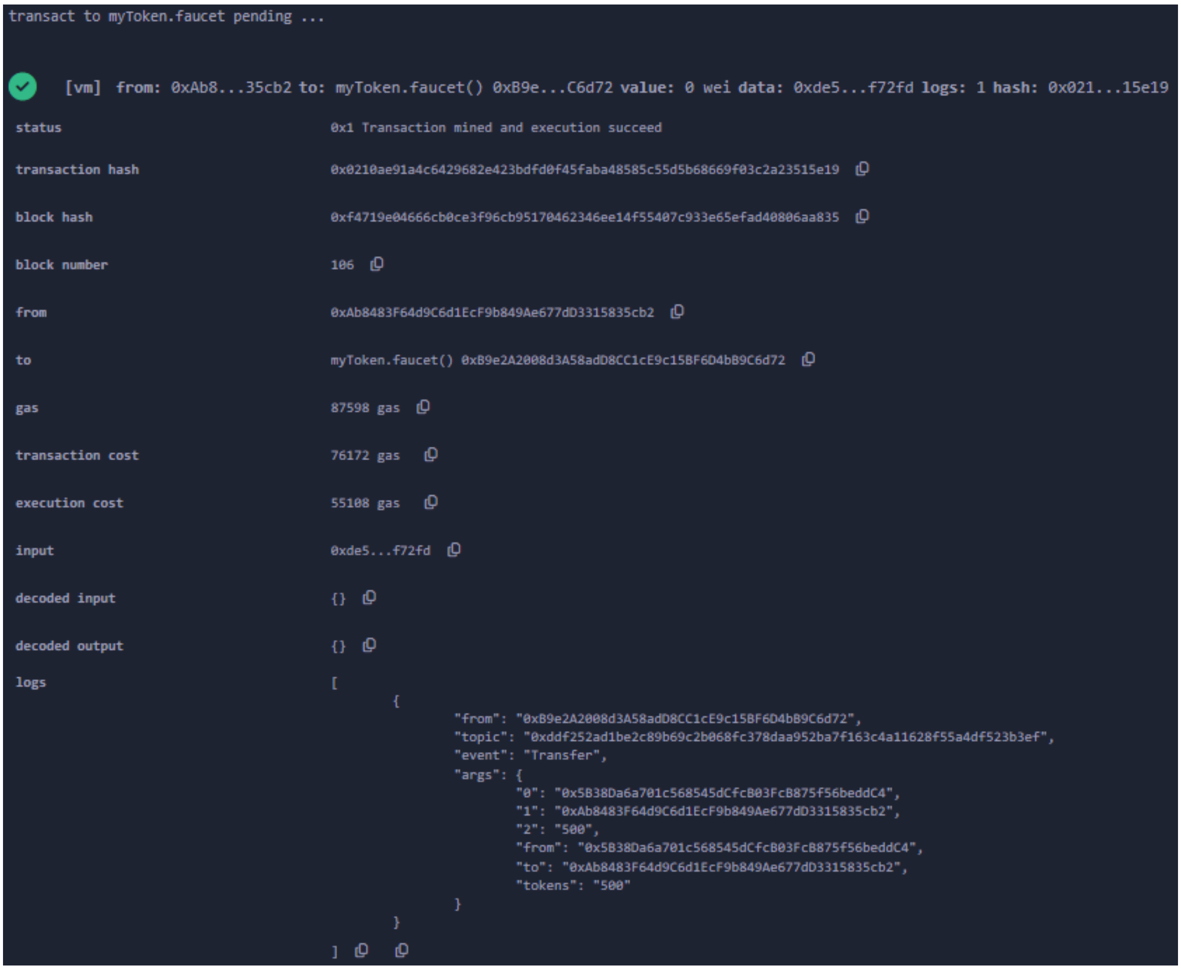
Status	View on block explorer
已確認	複製交易 ID
來源帳戶	目的帳戶
0x5cbcd...3...	0xF327C...

交易

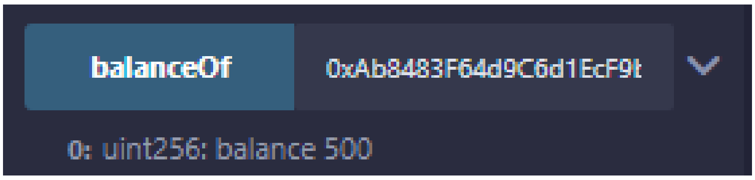
Nonce	1
數量	-0.1 DDOS
Gas 上限 (單位)	53244
Gas 用量 (單位)	52453
Base fee (GWEI)	0.000506324
Priority fee (GWEI)	1.5
Total gas fee	0.000079 SepoliaETH
Max fee per gas	0.000000002 SepoliaETH
總量	0.00007871 SepoliaETH

+ 活動紀錄

2. Write a function similar to the faucet, and you can add some variables for it. One screenshot is needed after the user using “faucet” function, and call “balanceOf” function.



```
transact to myToken.faucet pending ...

✓  [vm]  from: 0xAb8...35cb2 to: myToken.faucet() 0xB9e...C6d72 value: 0 wei data: 0xde5...f72fd logs: 1 hash: 0x021...15e19
status                       0x1 Transaction mined and execution succeed
transaction hash             0x0210ae91a4c6429682e423bdfd0f45faba48585c55d5b68669f03c2a23515e19
block hash                   0xf4719e04666cb0ce3f96cb95170462346ee14f55407c933e65efad40806aa835
block number                 106
from                         0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2
to                           myToken.faucet() 0xB9e2A2008d3A58adD8CC1cE9c15BF6D4bB9C6d72
gas                          87598 gas
transaction cost             76172 gas
execution cost               55108 gas
input                        0xde5...f72fd
decoded input                {}
decoded output               {}
logs                         [
                                     {
                                             "from": "0xB9e2A2008d3A58adD8CC1cE9c15BF6D4bB9C6d72",
                                             "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                             "event": "Transfer",
                                             "args": {
                                                     "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                     "1": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                     "2": "500",
                                                     "from": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                     "to": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                     "tokens": "500"
                                             }
                                     }
                             ]
```

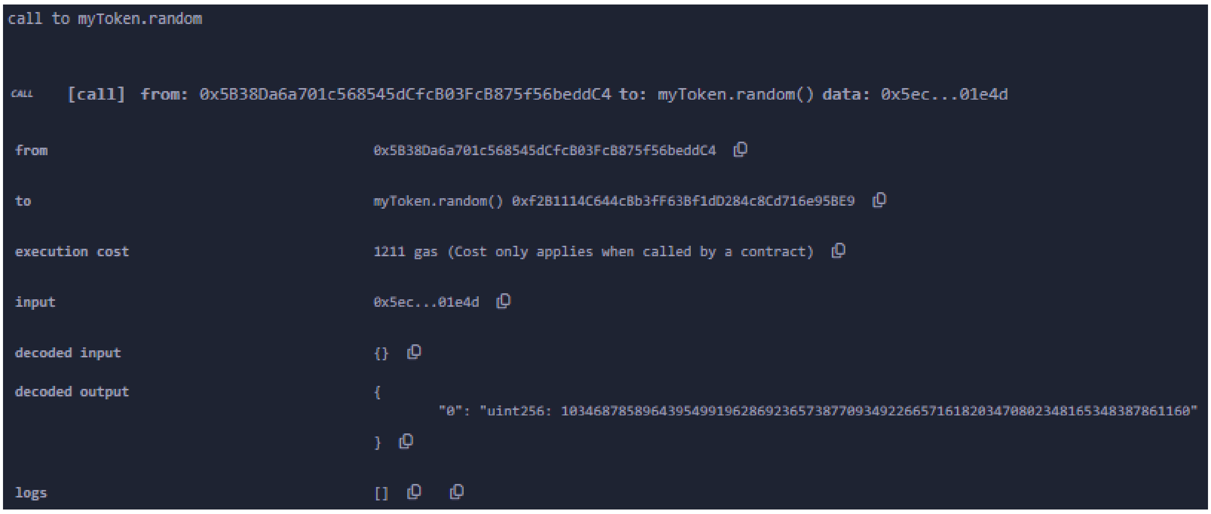


0xAb8483F64d9C6d1EcF9t

0: uint256: balance 500

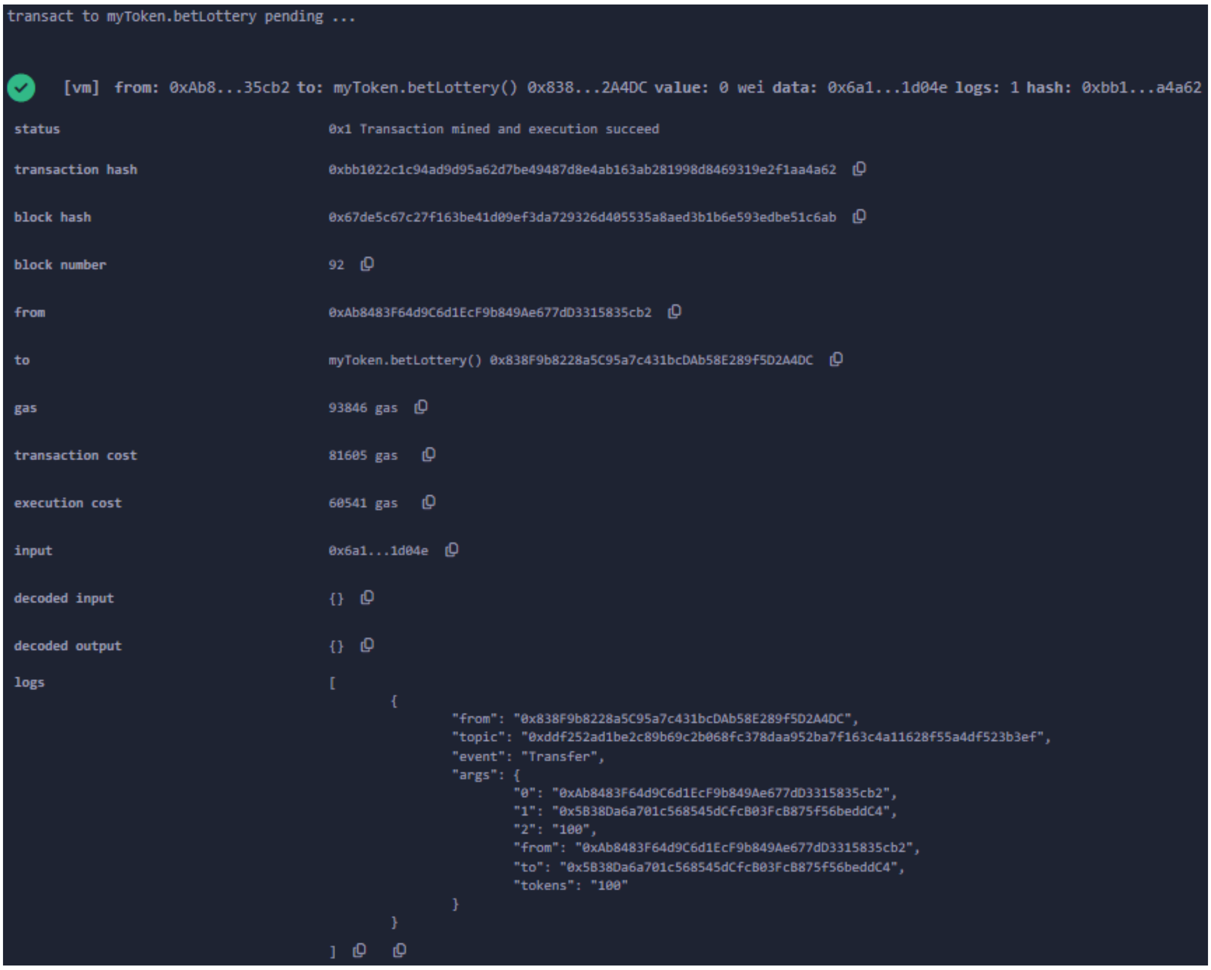
3. Complete a simple lottery game, including the following functions. Five screenshots(similar with e.g.) and function explanations(just for part B,C) are needed.

A. random

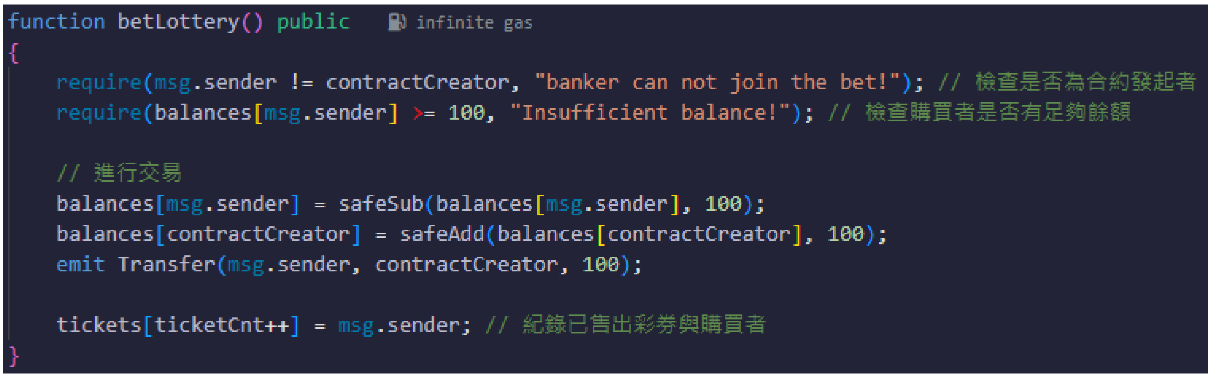


```
call to myToken.random


CALL   [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: myToken.random() data: 0x5ec...01e4d

from                        0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to                          myToken.random() 0xf2B1114C644cBb3fF63Bf1dD284c8Cd716e95BE9

execution cost              1211 gas (Cost only applies when called by a contract)

input                       0x5ec...01e4d

decoded input               {}

decoded output              {
                                    "0": "uint256: 103468785896439549919628692365738770934922665716182034708023481653483878611 60"
                            }

logs                        []
```

B. betLottery



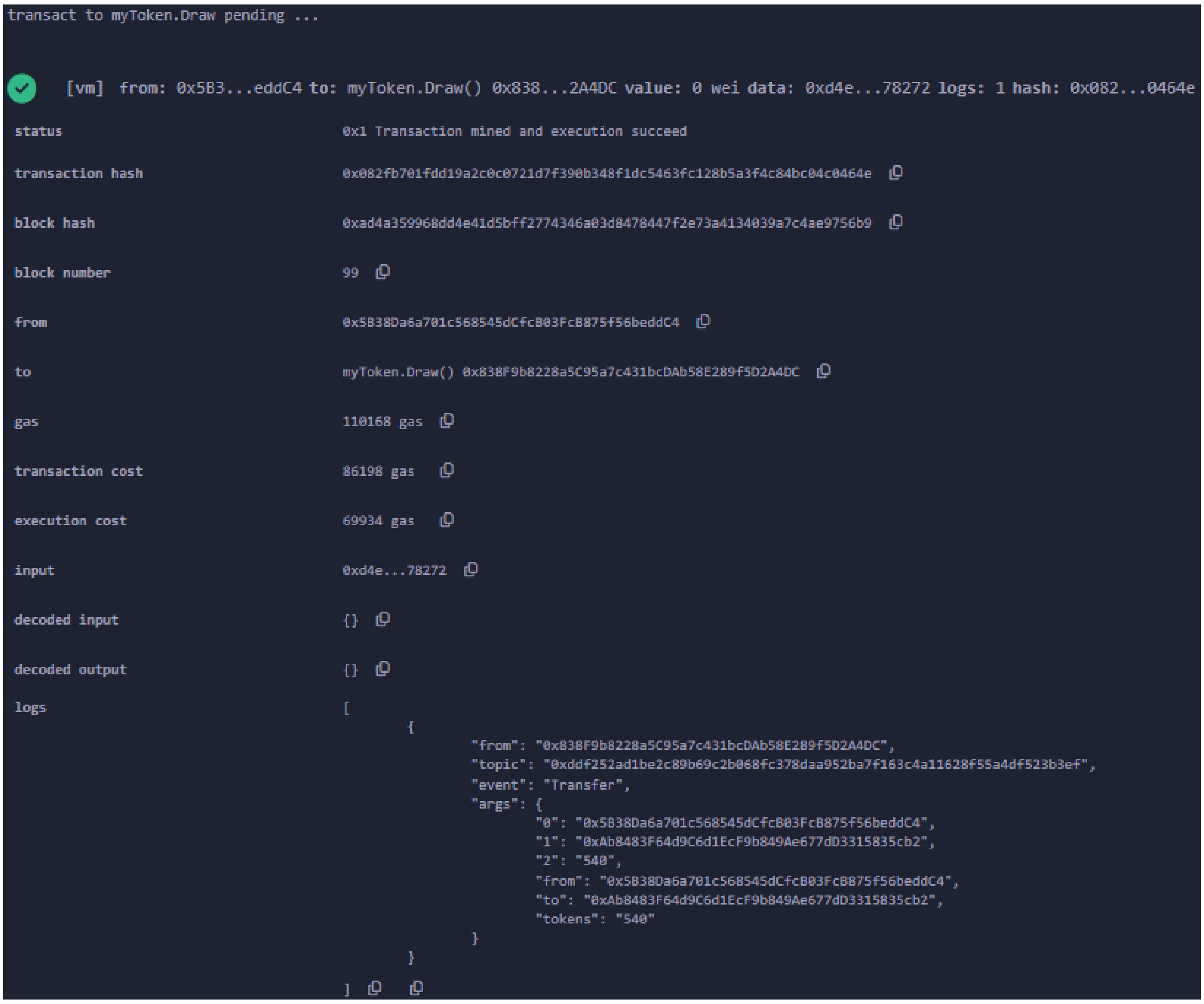
```
transact to myToken.betLottery pending ...


[vm] from: 0xAb8...35cb2 to: myToken.betLottery() 0x838...2A4DC value: 0 wei data: 0x6a1...1d04e logs: 1 hash: 0xbb1...a4a62
status                      0x1 Transaction mined and execution succeed

transaction hash            0xbb1022c1c94ad9d95a62d7be49487d8e4ab163ab281998d8469319e2f1aa4a62

block hash                  0x67de5c67c27f163be41d09ef3da729326d405535a8aed3b1b6e593edbe51c6ab

block number                92

from                        0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

to                          myToken.betLottery() 0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC

gas                         93846 gas

transaction cost            81605 gas

execution cost              60541 gas

input                       0x6a1...1d04e

decoded input               {}

decoded output              {}

logs                        [
                                    {
                                            "from": "0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC",
                                            "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                            "event": "Transfer",
                                            "args": {
                                                    "0": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                    "1": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                    "2": "100",
                                                    "from": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                    "to": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                    "tokens": "100"
                                            }
                                    }
                            ]
```



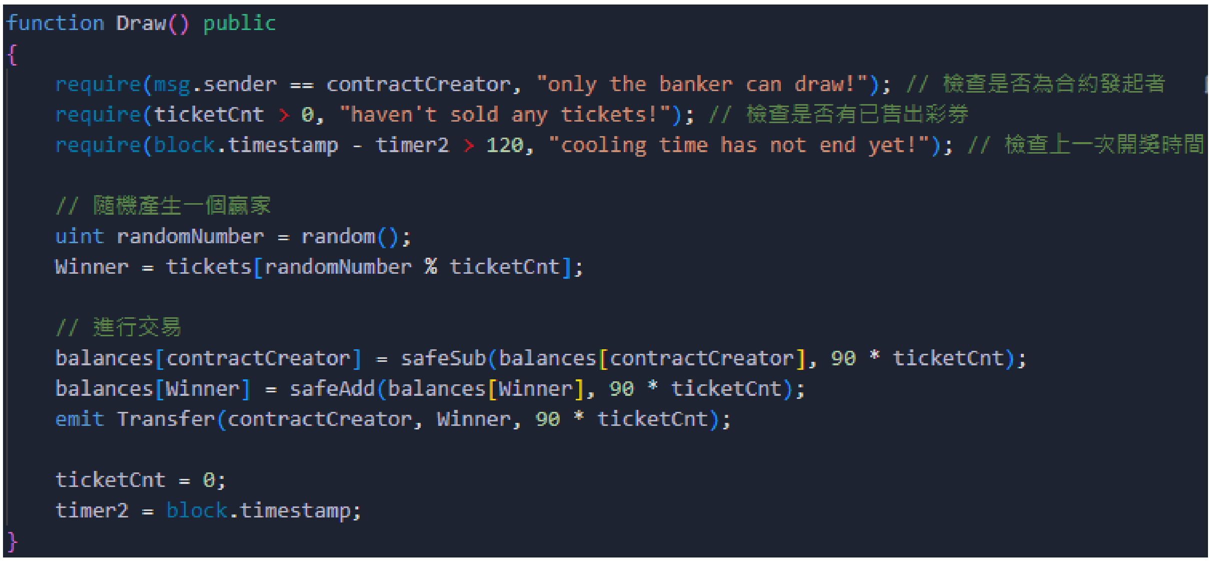
```
function betLottery() public    infinite gas
{
    require(msg.sender != contractCreator, "banker can not join the bet!"); // 檢查是否為合約發起者
    require(balances[msg.sender] >= 100, "Insufficient balance!"); // 檢查購買者是否有足夠餘額

    // 進行交易
    balances[msg.sender] = safeSub(balances[msg.sender], 100);
    balances[contractCreator] = safeAdd(balances[contractCreator], 100);
    emit Transfer(msg.sender, contractCreator, 100);

    tickets[ticketCnt++] = msg.sender; // 紀錄已售出彩券與購買者
}
```

首先檢查是否為合約發起者，合約發起者不能購買彩券，接著檢查購買者是否有足夠餘額，若有足夠餘額則交易成立，對雙方balances進行增減，發起Transfer，最後再紀錄已售出彩券與其購買者。

C. Draw



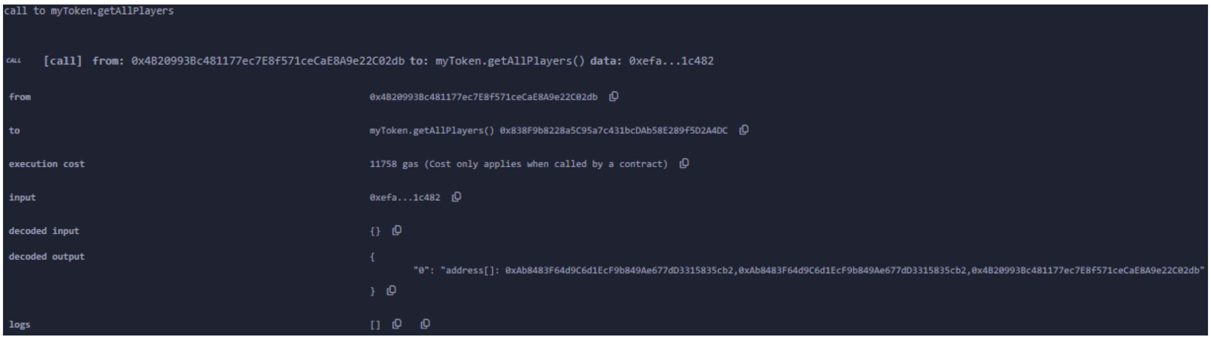
```
transact to myToken.Draw pending ...

✓  [vm]  from: 0x5B3...eddC4 to: myToken.Draw() 0x838...2A4DC value: 0 wei data: 0xd4e...78272 logs: 1 hash: 0x082...0464e
 status                    0x1 Transaction mined and execution succeed
 transaction hash          0x082fb701fdd19a2c0c0721d7f390b348f1dc5463fc128b5a3f4c84bc04c0464e
 block hash                0xad4a359968dd4e41d5bff2774346a03d8478447f2e73a4134039a7c4ae9756b9
 block number              99
 from                      0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
 to                        myToken.Draw() 0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC
 gas                       110168 gas
 transaction cost          86198 gas
 execution cost            69934 gas
 input                     0xd4e...78272
 decoded input             {}
 decoded output            {}
 logs                      [
                                   {
                                           "from": "0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC",
                                           "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
                                           "event": "Transfer",
                                           "args": {
                                                   "0": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                   "1": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                   "2": "540",
                                                   "from": "0x5B38Da6a701c568545dCfcB03FcB875f56beddC4",
                                                   "to": "0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2",
                                                   "tokens": "540"
                                           }
                                   }
                           ]
```



```
function Draw() public
{
    require(msg.sender == contractCreator, "only the banker can draw!"); // 檢查是否為合約發起者
    require(ticketCnt > 0, "haven't sold any tickets!"); // 檢查是否有已售出彩券
    require(block.timestamp - timer2 > 120, "cooling time has not end yet!"); // 檢查上一次開獎時間

    // 隨機產生一個贏家
    uint randomNumber = random();
    Winner = tickets[randomNumber % ticketCnt];

    // 進行交易
    balances[contractCreator] = safeSub(balances[contractCreator], 90 * ticketCnt);
    balances[Winner] = safeAdd(balances[Winner], 90 * ticketCnt);
    emit Transfer(contractCreator, Winner, 90 * ticketCnt);

    ticketCnt = 0;
    timer2 = block.timestamp;
}
```

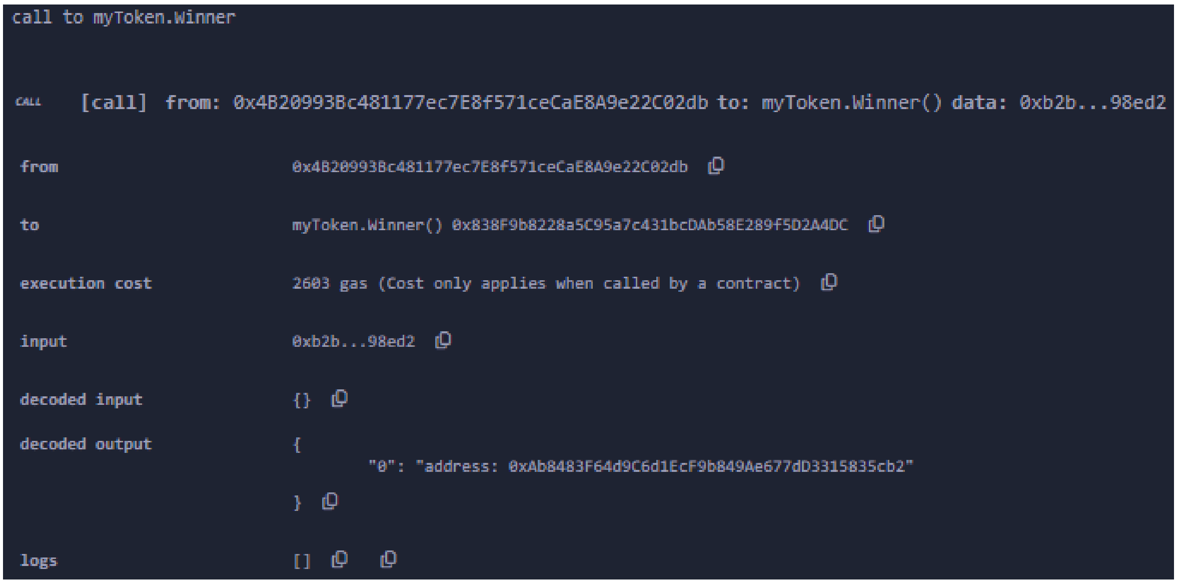
首先檢查是否為合約發起者，只有合約發起者能呼叫Draw function，接著檢查是否有已售出彩券，至少要有一張彩券售出才能進行抽獎，再檢查上一次開獎時間，需要至少相隔兩分鐘才能再次開獎。若都符合則呼叫random隨機產生一個Winner，進行交易，最後再重置紀錄的彩券與時間。

D. getAllPlayers



```
call to myToken.getAllPlayers

CALL   [call]  from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: myToken.getAllPlayers() data: 0xefa...1c482

from                 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
to                   myToken.getAllPlayers() 0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC
execution cost       11758 gas (Cost only applies when called by a contract)
input                0xefa...1c482
decoded input        {}
decoded output       {
                         "0": "address[]: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2,0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2,0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db"
                     }
logs                 []
```

E. Winner(just variable, not function)



```
call to myToken.Winner

CALL   [call]  from: 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db to: myToken.Winner() data: 0xb2b...98ed2

from                 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
to                   myToken.Winner() 0x838F9b8228a5C95a7c431bcDAb58E289f5D2A4DC
execution cost       2603 gas (Cost only applies when called by a contract)
input                0xb2b...98ed2
decoded input        {}
decoded output       {
                         "0": "address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2"
                     }
logs                 []
```