这是一个商品拍卖的智能合约。 有一个买家（获益的人） 和许多的竞价者， 竞价者对该商品进行竞价， 每次出价都会将相应的价值发送到智能合约中。 到拍卖结束的时候，
出价最高的人可以获得商品， 其余的人会拿回自己的钱。

据此，写出智能合约的 Solidity 代码：

```
pragma solidity ^0.4.24;

contract SimpleAuction{
    bytes32 public auctionlName;
    address public beneficiary;
    address public highestBidder;

    uint public auctionEnd;
    uint public highestOffer;11
    mapping(address=>uint) offers;
    bool ended;

    event HighestBidIncreased(address bidder,uint amount);
    event AuchtionEnded(address winner,uint amount);

    constructor(bytes32 _auctionlName, uint _biddingTime, address
_beneficiary) public{
        auctionlName = _auctionlName;
        beneficiary = beneficiary;
        auctionEnd = block.timestamp + _biddingTime;
    }
    function bid() public payable{
        require(block.timestamp <= auctionEnd);
        require(msg.value > highestOffer);

        if(highestBidder != 0){
            offers[highestBidder] += highestOffer;
        }
        highestBidder = msg.sender;
        highestOffer = msg.value;
        HighestBidIncreased(msg.sender,msg.value);
    }

    function withdraw() public view returns(bool){
        uint amount = offers[msg.sender];
        if (amount > 0){
            offers[msg.sender] = 0;
```

```
            if (!msg.sender.send(amount)){
                offers[msg.sender] = amount;
                return false;
            }
        }
        return true;
    }
    function auctionEnd() public{
        require(block.timestamp >= auctionEnd);
        require(!ended);
        ended = true;
        AuchtionEnded(highestBidder, highestOffer);
        beneficiary.transfer(highestOffer);
    }
}
```

相关接口的解释：

```
1.  function bid() public payable
```

竞价，当需要对拍卖商品出价时调用

```
2.  function withdraw() public view returns(bool)
```

当某人的出价被其他人超过时，他可以选择调用该函数来撤销他之前的出价

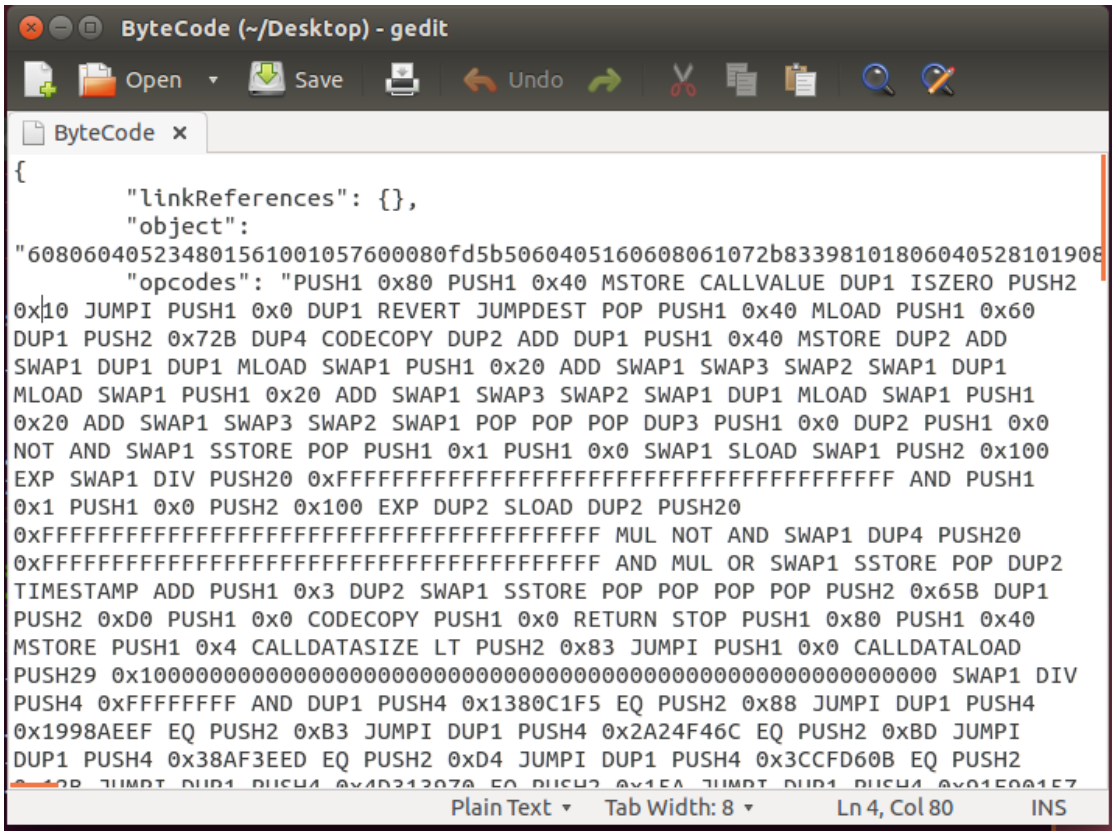
```
3.  function auctionEnd() public
```

当竞拍结束时，取消竞拍，将最高的出价交给拍卖品所有者

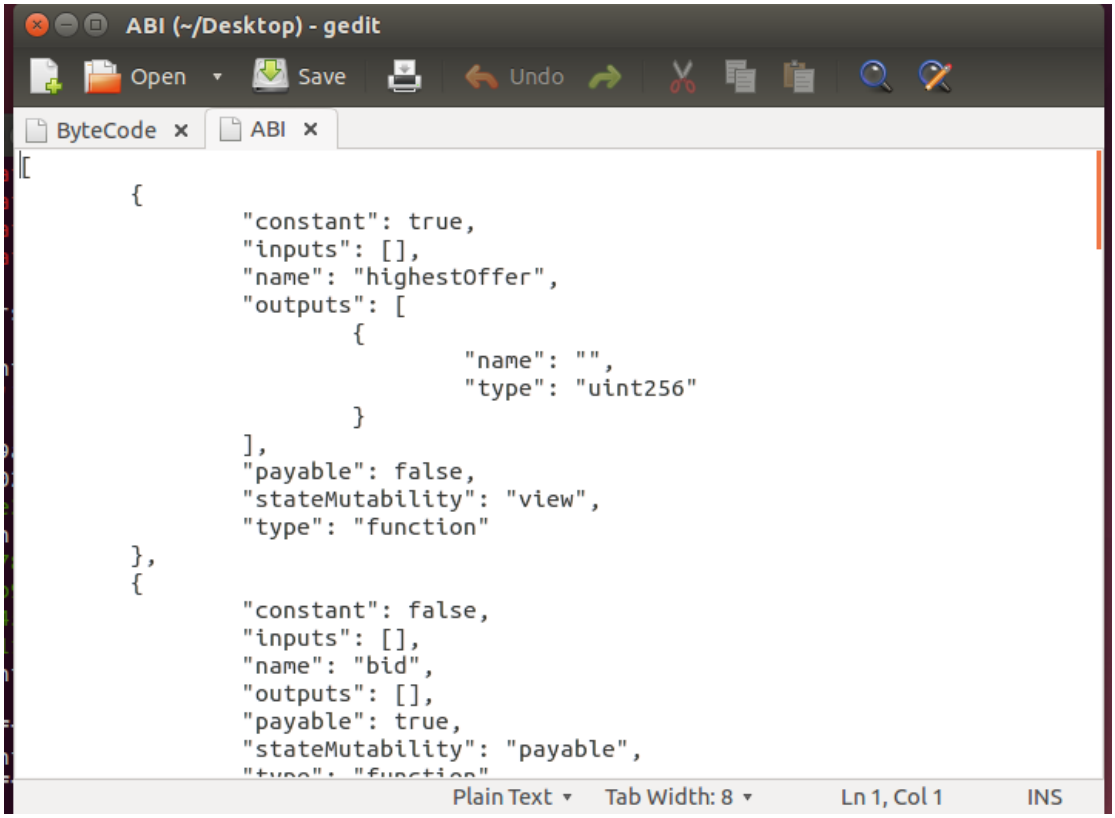
合约部署：

1. 通过 https://remix.ethereum.org，获取智能合约的 ABI 和字节码：

   字节码：



ABI：



2. 通过 ABI 创建一个合约：

命令：

```
> myContract = web3.eth.contract(abi)
```

结果：

```
> myContract
{
  abi: [{
      constant: true,
      inputs: [],
      name: "highestOffer",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "bid",
      outputs: [],
      payable: true,
      stateMutability: "payable",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "auctionEnd",
      outputs: [],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }, {
      constant: true,
      inputs: [],
      name: "beneficiary",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "withdraw",
      outputs: [{...}],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
```


type: "event"
}, {
anonymous: false,
inputs: [{...}, {...}],
name: "AuchtionEnded",
type: "event"
}],
eth: {
accounts: ["0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03", "0x1566f3a17a1b89bd
99a88f1ffe5bdcca12b9f8cb", "0x668a62d20c07b48d1d98d6099ba60d1a230470fe", "0x91d9
1640ea6a55df669cac45a64a576263a0b780", "0xab5610050281c73be3f8415cf3a5a49ec6d7cf
f0", "0x2481a4867811e6d2d8b96aecef0839bfb464f563"],
blockNumber: 202,
coinbase: "0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03",
compile: {
lll: function(),
serpent: function(),
solidity: function()
},
defaultAccount: undefined,
defaultBlock: "latest",
gasPrice: 1000000000,
hashrate: 0,
mining: false,
pendingTransactions: [{
blockHash: null,
blockNumber: null,
from: "0x78a851e5d57a3be6a72db2e4b4675f9b3f4bee03",
gas: 90000,
gasPrice: 1000000000,
hash: "0xae395a9aed678955f1a8628192a1d9518fc6150805885e3536de5d288195351
2",
input: "0x1998aeef",
nonce: 4,
r: "0x1155a05601f490324bd08c3cfc59858690b7fdf516c131e9492d49b75a64ebe9",

```
        transactionIndex: 0,
        v: "0x41",
        value: 1
    }],
    protocolVersion: "0x3f",
    syncing: false,
    call: function(),
    chainId: function(),
    contract: function(abi),
    estimateGas: function(),
    filter: function(options, callback, filterCreationErrorCallback),
    getAccounts: function(callback),
    getBalance: function(),
    getBlock: function(),
    getBlockNumber: function(callback),
    getBlockTransactionCount: function(),
    getBlockUncleCount: function(),
    getCode: function(),
    getCoinbase: function(callback),
    getCompilers: function(),
    getGasPrice: function(callback),
    getHashrate: function(callback),
    getMining: function(callback),
    getPendingTransactions: function(callback),
    getProtocolVersion: function(callback),
    getRawTransaction: function(),
    getRawTransactionFromBlock: function(),
    getStorageAt: function(),
    getSyncing: function(callback),
    getTransaction: function(),
    getTransactionCount: function(),
    getTransactionFromBlock: function(),
    getTransactionReceipt: function(),
    getUncle: function(),
    getWork: function(),
    iban: function(iban),
    icapNamereg: function(),
    isSyncing: function(callback),
    namereg: function(),
    resend: function(),
    sendIBANTransaction: function(),
    sendRawTransaction: function(),
    sendTransaction: function(),
    sendTransaction: function(),
    sign: function(),
    signTransaction: function(),
    submitTransaction: function(),
    submitWork: function()
  },
  at: function(address, callback),
  getData: function(),
  new: function()
}
```

3. 使用字节码预估手续费：

```
> web3.eth.estimateGas({data:bytecode})
527325
```

4. 部署合约：

```
> contractInstance = myContract.new({data: bytecode gas: 10000000, from: user1},
 function(e, contract){
.....    if(!e){
.........      if(!contract.address){
............        console.log("Contract transaction send: Transaction Hash: "+c
ontract.transactionHash+" waiting to be mined...");
............      }else{
............        console.log("Contract mined! Address: "+contract.address);
............        console.log(contract);
............      }
.........    }else{
.........      console.log(e)
.........    }
..... })
```

部署后输出如下：

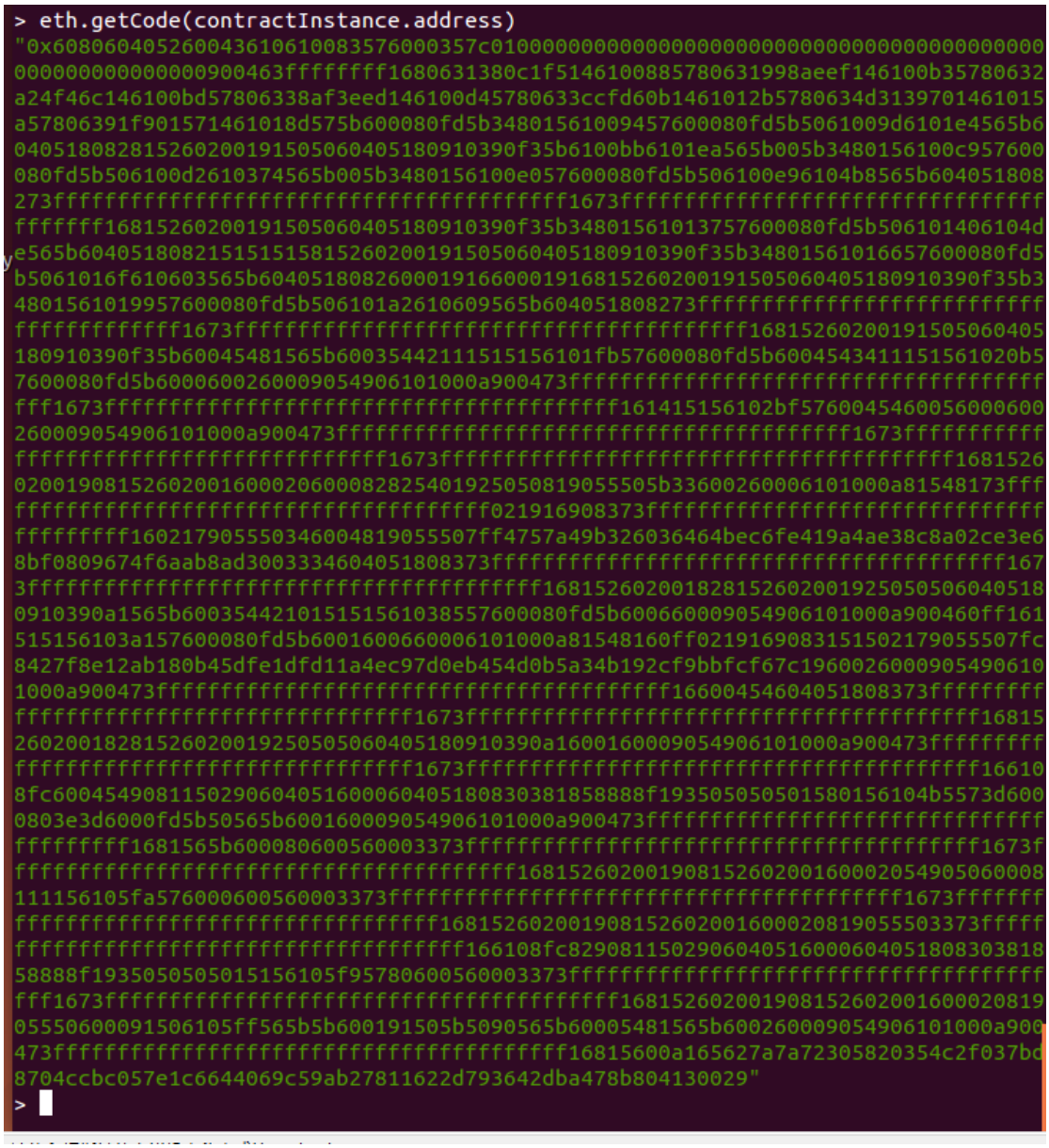
```
{
  abi: [{
      constant: true,
      inputs: [],
      name: "highestOffer",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "bid",
      outputs: [],
      payable: true,
      stateMutability: "payable",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "auctionEnd",
      outputs: [],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }, {
      constant: true,
      inputs: [],
      name: "beneficiary",
      outputs: [{...}],
      payable: false,
      stateMutability: "view",
      type: "function"
  }, {
      constant: false,
      inputs: [],
      name: "withdraw",
      outputs: [{...}],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }, {
      constant: true,
```

```
    inputs: [],
    name: "highestBidder",
    outputs: [{...}],
    payable: false,
    stateMutability: "view",
    type: "function"
  }, {
    inputs: [{...}, {...}, {...}],
    payable: false,
    stateMutability: "nonpayable",
    type: "constructor"
  }, {
    anonymous: false,
    inputs: [{...}, {...}],
    name: "HighestBidIncreased",
    type: "event"
  }, {
    anonymous: false,
    inputs: [{...}, {...}],
    name: "AuchtionEnded",
    type: "event"
  }],
  address: undefined,
  transactionHash: "0x85120bc2fe466c2cd791ede47e4f145c3c851df084c15ec9dba7f327c1
90ae6a"
}
```

合约开始等待挖矿。挖矿之后合约写入，：

```
INFO [11-27|02:57:04.724] Commit new mining work                   number=204 se
alhash=fbb489…186bbd uncles=0 txs=0 gas=0      fees=0           elapsed=30.783ms
Contract mined! Address: 0xf3eddd2391da2118c928b1de758073b13331afa9
```

可以查询到合约的地址：



合约已经具有地址，说明合约部署成功。