pragma solidity ^0.4.19;

contract BaseToken {

string public name;

string public symbol;

uint8 public decimals;

uint256 public totalSupply;

mapping (address => uint256) public balanceOf;

mapping (address => mapping (address => uint256)) public allowance;

event Transfer(address indexed from, address indexed to, uint256 value);

event Approval(address indexed owner, address indexed spender, uint256 value);

function \_transfer(address \_from, address \_to, uint \_value) internal {

require(\_to != 0x0);

require(balanceOf[\_from] >= \_value);

require(balanceOf[\_to] + \_value > balanceOf[\_to]);

uint previousBalances = balanceOf[\_from] + balanceOf[\_to];

balanceOf[\_from] -= \_value;

balanceOf[\_to] += \_value;

assert(balanceOf[\_from] + balanceOf[\_to] == previousBalances);

Transfer(\_from, \_to, \_value);

}

function transfer(address \_to, uint256 \_value) public returns (bool success) {

\_transfer(msg.sender, \_to, \_value);

return true;

}

function transferFrom(address \_from, address \_to, uint256 \_value) public returns (bool success) {

require(\_value <= allowance[\_from][msg.sender]);

allowance[\_from][msg.sender] -= \_value;

\_transfer(\_from, \_to, \_value);

return true;

}

function approve(address \_spender, uint256 \_value) public returns (bool success) {

allowance[msg.sender][\_spender] = \_value;

Approval(msg.sender, \_spender, \_value);

return true;

}

}

contract BurnToken is BaseToken {

event Burn(address indexed from, uint256 value);

function burn(uint256 \_value) public returns (bool success) {

require(balanceOf[msg.sender] >= \_value);

balanceOf[msg.sender] -= \_value;

totalSupply -= \_value;

Burn(msg.sender, \_value);

return true;

}

function burnFrom(address \_from, uint256 \_value) public returns (bool success) {

require(balanceOf[\_from] >= \_value);

require(\_value <= allowance[\_from][msg.sender]);

balanceOf[\_from] -= \_value;

allowance[\_from][msg.sender] -= \_value;

totalSupply -= \_value;

Burn(\_from, \_value);

return true;

}

}

contract CustomToken is BaseToken, BurnToken {

function CustomToken() public {

totalSupply = 8000000000000000;

balanceOf[0x1dd91123acc8a51392b35b310b2f0bed6ff082f2] = totalSupply;

name = 'Teamothers';

symbol = 'TOS';

decimals = 8;

}

}