// R Coins ICO

pragma solidity ^0.4.25;

contract rcoin\_ico {

// Introducing the maximum number of Rcoins available for sale

uint public max\_rcoins = 1000000;

// Introducing the USD to Rcoins conversion rate

uint public usd\_to\_rcoins = 1000;

// Introducing the total number of Rcoins that have been bought by the investors

uint public total\_rcoins\_bought = 0;

// Mapping from the investor address to its equity in Rcoins and USD

mapping(address => uint) equity\_rcoins;

mapping(address => uint) equity\_usd;

// Checking if an investor can buy Rcoins

modifier can\_buy\_rcoins(uint usd\_invested) {

require (usd\_invested \* usd\_to\_rcoins + total\_rcoins\_bought <= max\_rcoins);

\_; // the underscore only does the modifier if the condition is true

}

// Getting the equity in Rcoins of an investor

function equity\_in\_rcoins(address investor) external constant returns (uint) {

return equity\_rcoins[investor];

}

// Getting the equity in USD of an investor

function equity\_in\_usd(address investor) external constant returns (uint) {

return equity\_usd[investor];

}

// Buying Rcoins

function buy\_rcoins(address investor, uint usd\_invested) external

can\_buy\_rcoins(usd\_invested) {

uint rcoins\_bought = usd\_invested \* usd\_to\_rcoins;

equity\_rcoins[investor] += rcoins\_bought;

equity\_usd[investor] = equity\_rcoins[investor] / 1000;

total\_rcoins\_bought += rcoins\_bought;

}

// Selling Rcoins

function sell\_rcoins(address investor, uint rcoins\_toSell) external {

equity\_rcoins[investor] -= rcoins\_toSell;

equity\_usd[investor] = equity\_rcoins[investor] / 1000;

total\_rcoins\_bought -= rcoins\_toSell;

}

}