Day17

pragma solidity ^0.4.25;

library Set {

struct Data {

mapping(int => bool) data;

}

function Insert(Data storage self, int key) public returns (bool) {

if (self.data[key])

return false; // Key exists.

self.data[key] = true;

return true;

}

function Remove(Data storage self, int key) public returns (bool) {

if (!self.data[key])

return false; // Key does not exist.

self.data[key] = false;

return true;

}

function Contain(Data storage self, int key) public view returns (bool) {

return self.data[key];

}

}

contract Main {

Set.Data set;

function insert(int key) public returns (bool) {

return Set.Insert(set, key);

}

function remove(int key) public returns (bool) {

return Set.Remove(set, key);

}

function contain(int key) public view returns (bool) {

return Set.Contain(set, key);

}

}