pragma solidity ^0.5.1;

contract Hospital\_Trainee\_Competency {

address public \_NationalAuthority;

struct Hospital {

address \_HospitalAddress;

uint \_HospitalID;

string \_HospitalName;

}

Hospital H;

mapping (address => Hospital) Hos;

Hospital[] public ListOfHospital;

struct Trainee {

uint \_TraineeID;

string \_TraineeName;

string \_TraineeCompetency;

}

Trainee T;

mapping (address => Trainee) Tra;

Trainee[] public ListOfTrainee;

constructor () public {

\_NationalAuthority = msg.sender;

}

function Authorise\_Hospital(address \_HospitalAddress, uint \_HospitalID, string memory \_HospitalName) public {

require(msg.sender == \_NationalAuthority);

Hospital storage hos = Hos[\_HospitalAddress];

hos.\_HospitalAddress = \_HospitalAddress;

hos.\_HospitalID = \_HospitalID;

hos.\_HospitalName = \_HospitalName;

ListOfHospital.push(hos);

}

function See\_Authorised\_Hospital(address \_HospitalAddress) view public returns (address, uint, string memory){

return (Hos[\_HospitalAddress].\_HospitalAddress, Hos[\_HospitalAddress].\_HospitalID, Hos[\_HospitalAddress].\_HospitalName);

}

}