// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.0;

   contract HOSP{

    address public Patient1;

     constructor() {

         Patient1= address(this);

    }

   struct Doctor

   {

      string Dname;

       string qual;

      string clinic;

   }

 mapping(uint=>Doctor) SDoctors;

   function addDoctor(uint key, string memory \_Dname, string memory \_qual,string memory \_clinic)

     public{ SDoctors[key]=Doctor(\_Dname,\_qual,\_clinic);

   }

   function querybyid(uint \_id) public view returns(

        string memory \_Dname,string memory \_qual,string memory \_clinic)

        {

            return(SDoctors[\_id].Dname,SDoctors[\_id].qual,SDoctors[\_id].clinic);

      }

struct Patient

   {

      string Pname;

      uint age;

      string disease;

   }

 mapping(uint=>Patient) SPatients;

   function addPatient(uint key, string memory \_Pname, uint \_age, string memory \_disease)

     public{ SPatients[key]=Patient(\_Pname,\_age,\_disease);

   }

   function querybyPid(uint \_id) public view returns(

        string memory \_Pname,uint \_age,string memory \_disease)

        {

            return(SPatients[\_id].Pname,SPatients[\_id].age,SPatients[\_id].disease);

      }

        struct Medicine

   {

      uint Mid;

      string Mname;

      string DOE;

      string Dose;

      uint Price;

   }

   mapping(uint=>Medicine)SMedicine;

   function addmedicine(uint key, uint \_Mid, string memory  \_Mname, string memory \_DOE,string memory \_Dose, uint  \_Price)

         public{SMedicine[key]=Medicine(\_Mid,\_Mname,\_DOE,\_Dose, \_Price);

            }

   function querybyMid(uint \_id) public view returns(uint \_Mid,string memory \_Mname, string memory \_DOE,string memory \_Dose, uint \_Price)

        {

   return(SMedicine[\_id].Mid,SMedicine[\_id].Mname,SMedicine[\_id].DOE, SMedicine[\_id].Dose,SMedicine[\_id].Price);

         }

}