// elections project

// SPDX-License-Identifier: GPL-3.0

pragma solidity>0.5.0<0.9.0;

/\*

task description: three party will contest election and only one party will.

Final result wuill declare after all n members voted.

\*/

contract evoting

{

    //variables

    string  bjp; // its just memory allocation, nothing stored.

    string congress;

    string left;

    uint public CountBjp=0;

    uint public CountCong=0;

    uint public CountLeft=0;

    mapping(address=>bool)  public voters;

    uint public totalVoters=0;

    uint public realVotedNo;

    bool public deadline=false;

    function voterRegistration(address voterId) public

    {

        require(deadline==false);

        require( voters[voterId]!=true);

        voters[voterId]=true;

        realVotedNo=realVotedNo+1;

    }

    function  VotingdayStart() public returns(string memory)

    {

        deadline=true;

        return ("deadline passed vo more political activities");

    }

    function voting(address voterName, string memory party) public returns(string memory)

    {

        require(voters[voterName]==true  && deadline==true);

        voters[voterName]=false;

        if (keccak256(abi.encodePacked(party)) == keccak256(abi.encodePacked('bjp')))

        {CountBjp=CountBjp+1;

         }

        if (keccak256(abi.encodePacked(party)) == keccak256(abi.encodePacked('congress')))

        {CountCong=CountCong+1;

        }

        if (keccak256(abi.encodePacked(party)) == keccak256(abi.encodePacked('left')))

        {CountLeft=CountLeft+1;

        }

        totalVoters=totalVoters+1;

    }

    function winner() view public returns(string memory)

    {

        uint  max=0;

        string memory party;

        if(CountCong>max)

        {

            max=CountCong;

            party= 'congress';

        }

         if(CountBjp>max)

        {

            max=CountBjp;

            party= 'bjp';

        }

        if(CountLeft>max)

        {

            max=CountLeft;

            party= 'left';

        }

        return(party);

    }

}