```
pragma solidity^0.4.24;

contract MyContract {
    string value;

    constructor() public {
        value= "myValue";

    }
    function get() public view returns(string)
    {
        return value;

    }
    function set(string _value) public {
        value= _value;

    }
}
```

```
pragma solidity ^0.8.19;
contract dataPerson{
    uint256 public peopleCount=0;
    mapping(uint=>Person) public people;

    struct Person{
        uint _id;
        string _firstname;
        string _lastname;
    }
    function addPerson(string memory _firstname, string memory _lastname)
        incrementCount();
        people[peopleCount]=Person(peopleCount,_firstname,_lastname);
    }
    function incrementCount() internal {
        peopleCount += 1;
    }
}
```

```
pragma solidity 0.5.1;

contract counter {
uint public Count=0;
event Increment(uint value);
event Decrement(uint value);

function getCount() view public returns(uint){
return Count;
}
function increment() public{
    Count += 1;
    emit Increment(Count);
}
function decrement() public {
    Count -= 1;
    emit Decrement(Count);
}
}
```

```
pragma solidity ^0.8.1;
contract dataPerson{
    uint256 public peopleCount=0;
    mapping(uint=>Person) public people;

    struct Person{
        uint _id;
        string _firstname;
        string _lastname;
    }
    function addPerson(string memory _firstname, string memory _lastname)
        incrementCount();
        people[peopleCount]=Person(peopleCount,_firstname,_lastname);
    }
    function incrementCount() internal {
        peopleCount += 1;
    }
}
```