

1. Write , compile and show the deployment output of a smart contract with the following functions-

a. A function to perform pre increment operation on an uint256 variable.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Counter{
    uint256 count =1;
    int256 count1=1;
    function uintPreIncrement() public returns (uint256){
        return(++count);
    }
    function intPostIncrementInt() public returns (int256){
        return(count1++);
    }
    function intPreDecrementInt() public returns (int256){
        return(--count1);
    }
    function uintPostDecrementInt() public returns (uint256){
        return(count--);
    }
}
```

Transactions recorded: 19

Deployed Contracts: COUNTER AT 0x7B9

Low level interactions: CALLDATA

input: 0xa0...79383

decoded input: {}

decoded output: { "0": "int256: 1" }

logs: []

val: 0 wei

b. A function to perform post increment operation on an int256 variable.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Counter{
    uint256 count =1;
    int256 count1=1;
    function uintPreIncrement() public returns (uint256){
        return(++count);
    }
    function intPostIncrementInt() public returns (int256){
        return(count1++);
    }
    function intPreDecrementInt() public returns (int256){
        return(--count1);
    }
    function uintPostDecrementInt() public returns (uint256){
        return(count--);
    }
}
```

Transactions recorded: 20

Deployed Contracts: COUNTER AT 0x7B9

Low level interactions: CALLDATA

input: 0x02...cd1b7

decoded input: {}

decoded output: { "0": "int256: 1" }

logs: []

val: 0 wei

c. A function to perform pre decrement operation on an int256 variable.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Counter{
    uint256 count =1;
    int256 count1=1;
    function uintPreIncrement() public returns (uint256){
        return(++count);
    }
    function intPostIncrementInt() public returns (int256){
        return(count1++);
    }
    function intPreDecrementInt() public returns (int256){
        return(--count1);
    }
    function uintPostDecrementInt() public returns (uint256){
        return(count--);
    }
}
```

Transactions recorded: 21

Deployed Contracts: COUNTER AT 0x7B9

Low level interactions: CALLDATA

input: 0xe6b...a3742

decoded input: {}

decoded output: { "0": "uint256: 1" }

logs: []

val: 0 wei

Copy

d. A function to perform post decrement operation on an uint256 variable.

DEPLOY & RUN TRANSACTIONS

Home functions.sol assignment11.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Counter{
    uint256 count =1;
    int256 count1=1;
    function uintPreIncrement() public returns (uint256){
        return(++count);
    }
    function intPostIncrementInt() public returns (int256){
        return(count1++);
    }
    function intPreDecrementInt() public returns (int256){
        return(--count1);
    }
    function uintPostDecrementInt() public returns (uint256){
        return(count--);
    }
}
```

Transactions recorded 22

Deployed Contracts COUNTER AT 0x7B9

Low level interactions CALDATA

input 0x805...fe92e

decoded input {}

decoded output { "0": "uint256: 1"}

logs []

val 0 wei

Type here to search

17:49 ENG 23-04-2022