

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.

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, showing a list of deployed contracts for 'COUNTER AT 0x7B9'. Four buttons are visible: 'intPostIncr...', 'intPreDecr...', 'uintPostDecr...', and 'uintPreIncr...'. The 'uintPreIncr...' button is highlighted. The main editor displays the Solidity code for 'assignment11.sol':`1 //SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract Counter{
5 uint256 count =1;
6 int256 count1=1;
7 function uintPreIncrement() public returns (uint256){
8 return(++count);
9 }
10 function intPostIncrementInt() public returns (int256){
11 return(count1++);
12 }
13 function intPreDecrementInt() public returns (int256){
14 return(--count1);
15 }
16 function uintPostDecrementInt() public returns (uint256){
17 return(count--);
18 }
19 }`

The 'Run' tab at the bottom shows the execution details for the 'uintPreIncrement()' function. The input is an empty array, and the decoded output is '0: "int256: 1"'. The logs and val sections are empty.

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

The screenshot shows the Remix IDE interface. The 'uintPreIncr...' button is highlighted in the left panel. The main editor displays the same Solidity code as in the previous screenshot. The 'Run' tab shows the execution details for the 'intPostIncrementInt()' function. The input is an empty array, and the decoded output is '0: "int256: 1"'. The logs and val sections are empty.

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

The screenshot shows the Remix IDE interface. The 'intPreDecr...' button is highlighted in the left panel. The main editor displays the same Solidity code as in the previous screenshots. The 'Run' tab shows the execution details for the 'intPreDecrementInt()' function. The input is an empty array, and the decoded output is '0: "uint256: 1"'. The logs section is empty, and the val section shows '0 wei'. A 'Copy' button is visible next to the logs.

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

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' sidebar shows deployment options and a list of deployed contracts, including 'COUNTER AT 0x7B9'. The central editor shows the Solidity code for a 'Counter' contract. The right sidebar displays transaction logs and input/output data for the selected function.

```
1 //SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract Counter{
5     uint256 count =1;
6     int256 count1=1;
7     function uintPreIncrement() public returns (uint256){
8         return(++count);
9     }
10    function intPostIncrementInt() public returns (int256){
11        return(count1++);
12    }
13    function intPreDecrementInt() public returns (int256){
14        return(--count1);
15    }
16    function uintPostDecrementInt() public returns (uint256){
17        return(count--);
18    }
19 }
```

The right sidebar shows the following data:

Input	0x085...fe92e
decoded input	{}
decoded output	{ "0": "uint256: 1" }
logs	[]
val	0 wei