ASSIGNMENT DAY-5

1.How to work with Error Handling in Solidity ? (Require & Assert)

Ans: Solidity has many functions for error handling. Errors can occur at compile time or runtime. Solidity is compiled to byte code and there a syntax error check happens at compile-time, while runtime errors are difficult to catch and occurs mainly while executing the contracts. Some of the runtime errors are out-of-gas error, data type overflow error, divide by zero error, array-out-of-index error, etc. Until version 4.10 a single throw statement was there in solidity to handle errors, so to handle errors multiple if…else statements, one has to implement for checking the values and throw errors which consume more gas. After version 4.10 new error handling construct **assert, require, revert** statements were introduced and the throw was made absolute.

require() is used:

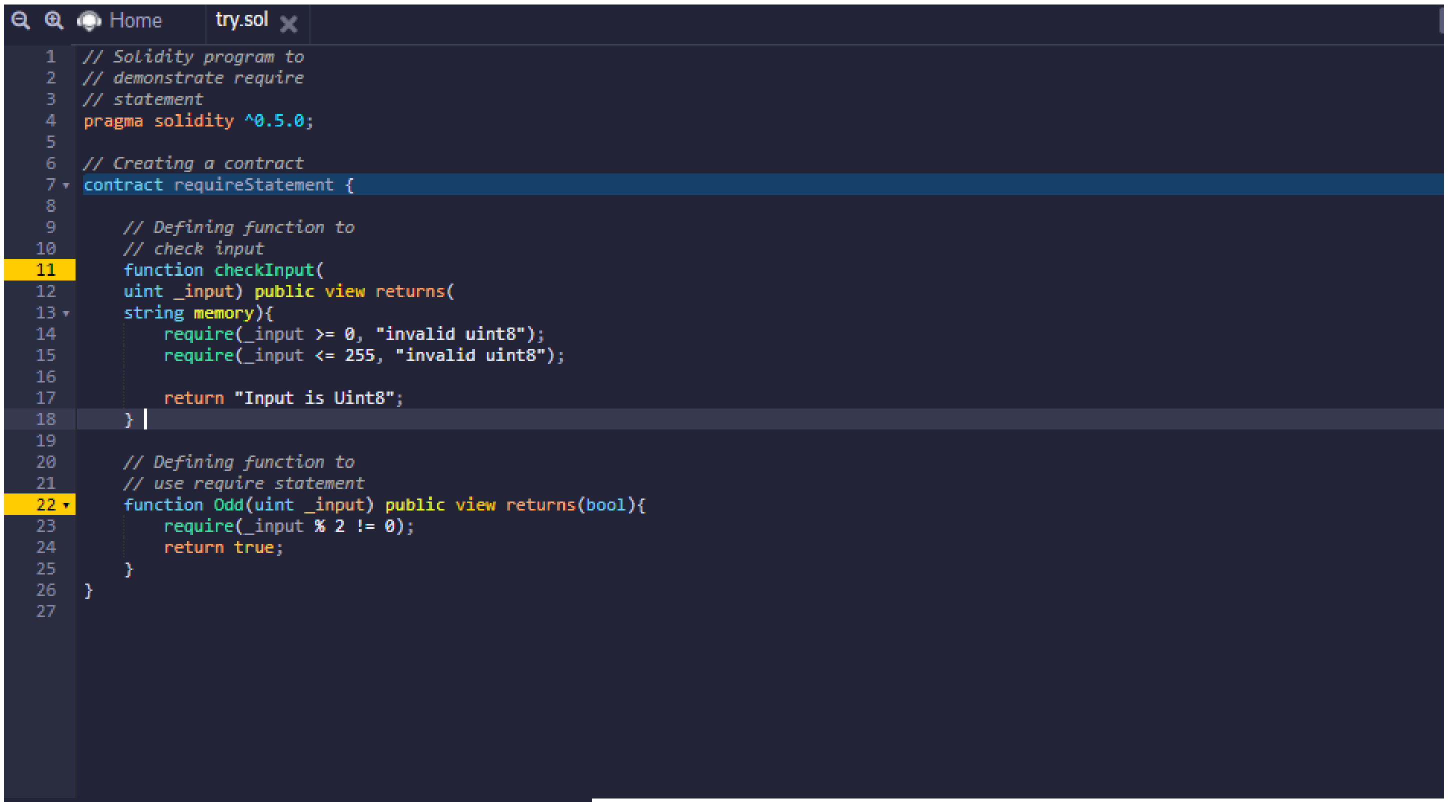
1.Validate user inputs.

2.Validate the response from an external contract.i.e.require(external.send(amount)).

3.Validate the state condition prior to execution.

4.Generally it is used at the beginning of a function.

Example of require():



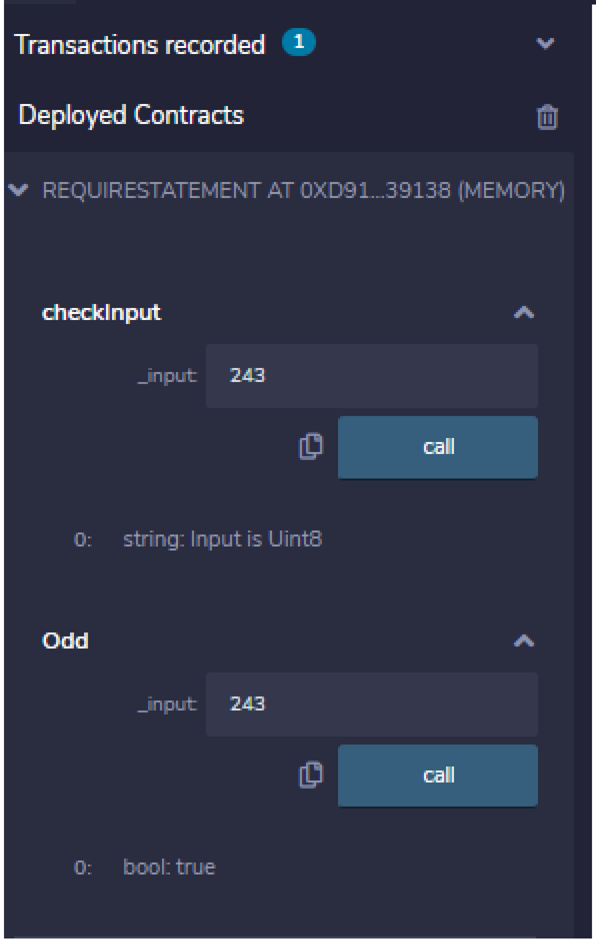
```
// Solidity program to
// demonstrate require
// statement
pragma solidity ^0.5.0;

// Creating a contract
contract requireStatement {

    // Defining function to
    // check input
    function checkInput(
    uint _input) public view returns(
    string memory){
        require(_input >= 0, "invalid uint8");
        require(_input <= 255, "invalid uint8");

        return "Input is Uint8";
    }

    // Defining function to
    // use require statement
    function Odd(uint _input) public view returns(bool){
        require(_input % 2 != 0);
        return true;
    }
}
```



```
checkInput  _input: 243
0: string: Input is Uint8

Odd  _input: 243
0: bool: true
```

assert() is used to:

1.Check for overflow/underflow.i.e. c=a*b; assert(c > b)

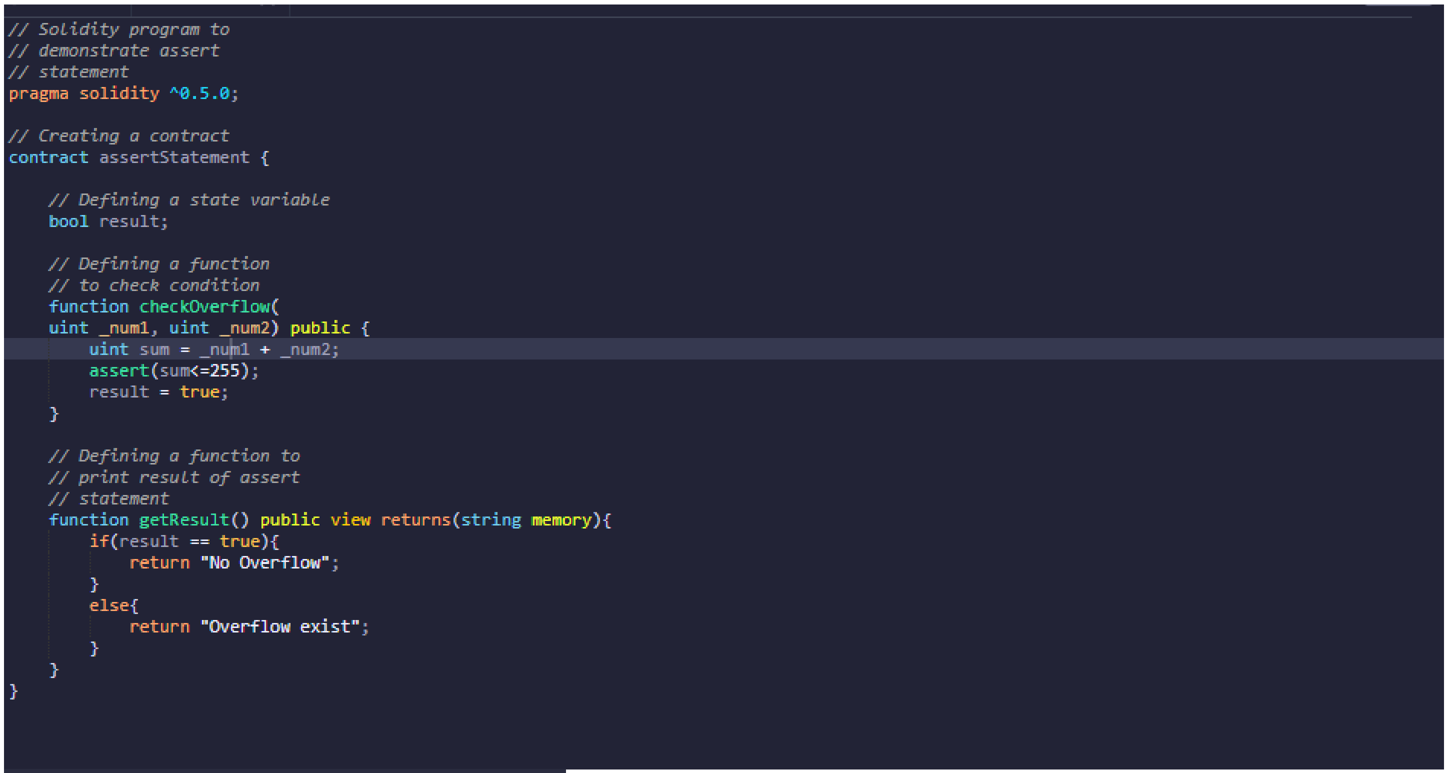
2.Check invariants. i.e assert(this.balance > totalBalance).

3.Validate state after making changes.

4.Prevent condition which should never ever be possible.

5.Generally it is used towards the end of a function.

Example of assert():



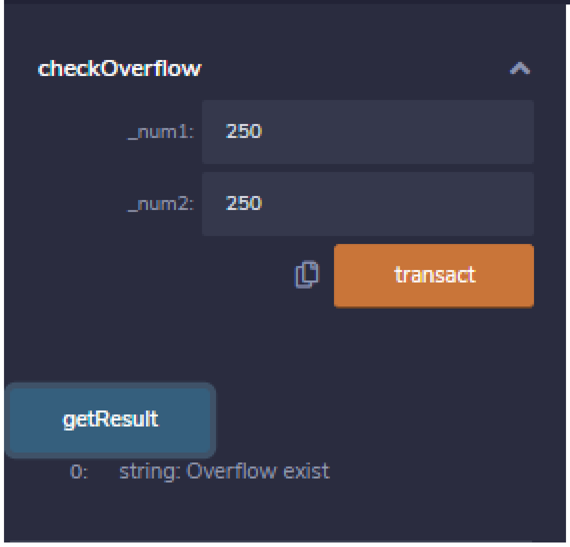
```
// Solidity program to
// demonstrate assert
// statement
pragma solidity ^0.5.0;

// Creating a contract
contract assertStatement {

    // Defining a state variable
    bool result;

    // Defining a function
    // to check condition
    function checkOverflow(
    uint _num1, uint _num2) public {
        uint sum = _num1 + _num2;
        assert(sum<=255);
        result = true;
    }

    // Defining a function to
    // print result of assert
    // statement
    function getResult() public view returns(string memory){
        if(result == true){
            return "No Overflow";
        }
        else{
            return "Overflow exist";
        }
    }
}
```



```
_num1: 250
_num2: 250

0:   string: Overflow exist
```