

SNAPSHOTS

(Student Name: Prakhar Sharma (NPCII))

1. Deploy Contract: Contract deployed with initial supply of 2000 Ethers.

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying the 'Deployed Contracts' section. The contract 'ERC20Token' is listed with a balance of 0 ETH. The 'approve' function is selected, and the 'totalSupply' is shown as 2000. The main editor displays the Solidity code for the ERC20Token contract, including the constructor, transfer, and approve functions. The console shows the execution of the constructor, which sets the initial supply to 2000.

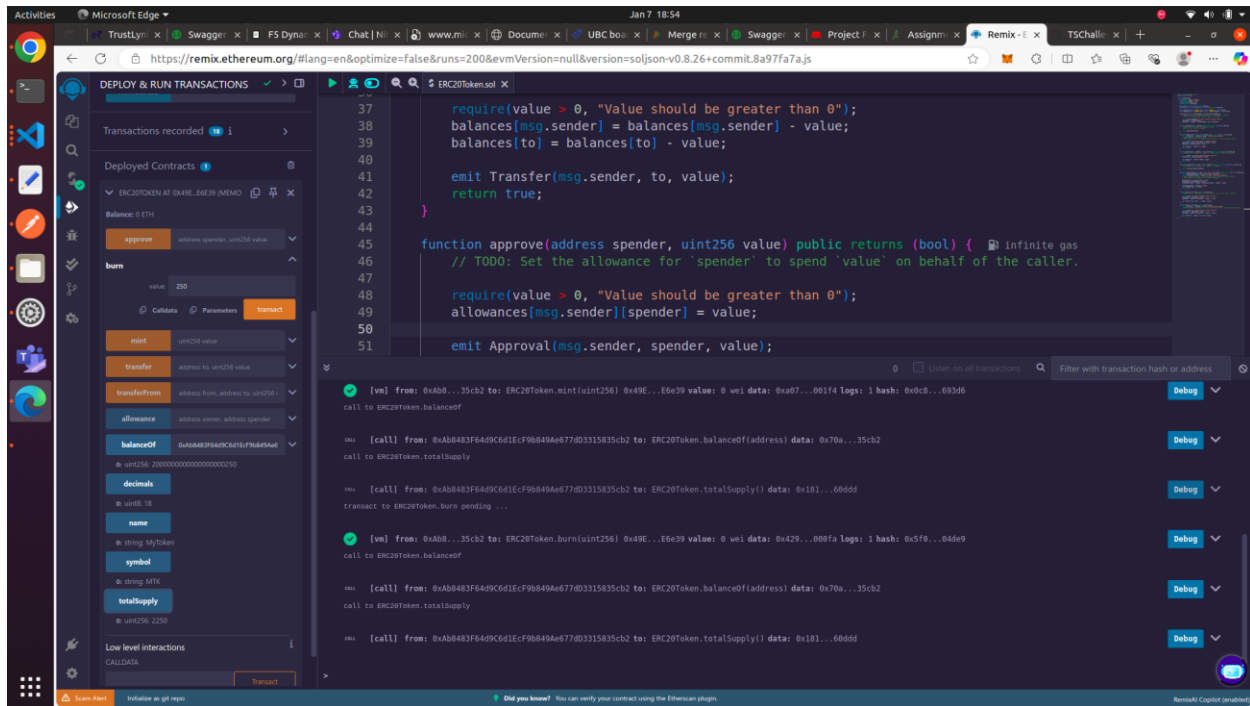
```
37 require(value > 0, "Value should be greater than 0");
38 balances[msg.sender] = balances[msg.sender] - value;
39 balances[to] = balances[to] + value;
40
41 emit Transfer(msg.sender, to, value);
42 return true;
43
44
45 function approve(address spender, uint256 value) public returns (bool) {
46 // TODO: Set the allowance for 'spender' to spend 'value' on behalf of the caller.
47
48 require(value > 0, "Value should be greater than 0");
49 allowances[msg.sender][spender] = value;
50
51 emit Approval(msg.sender, spender, value);
```

2. Mint: Minting 500 and now totalSupply is 2500.

The screenshot shows the Remix IDE interface after minting 500 tokens. The 'DEPLOY & RUN TRANSACTIONS' panel is active, and the 'mint' function is selected. The 'totalSupply' is now 2500. The main editor displays the Solidity code for the ERC20Token contract, including the constructor, transfer, and approve functions. The console shows the execution of the mint function, which increases the total supply to 2500.

```
37 require(value > 0, "Value should be greater than 0");
38 balances[msg.sender] = balances[msg.sender] - value;
39 balances[to] = balances[to] + value;
40
41 emit Transfer(msg.sender, to, value);
42 return true;
43
44
45 function approve(address spender, uint256 value) public returns (bool) {
46 // TODO: Set the allowance for 'spender' to spend 'value' on behalf of the caller.
47
48 require(value > 0, "Value should be greater than 0");
49 allowances[msg.sender][spender] = value;
50
51 emit Approval(msg.sender, spender, value);
```

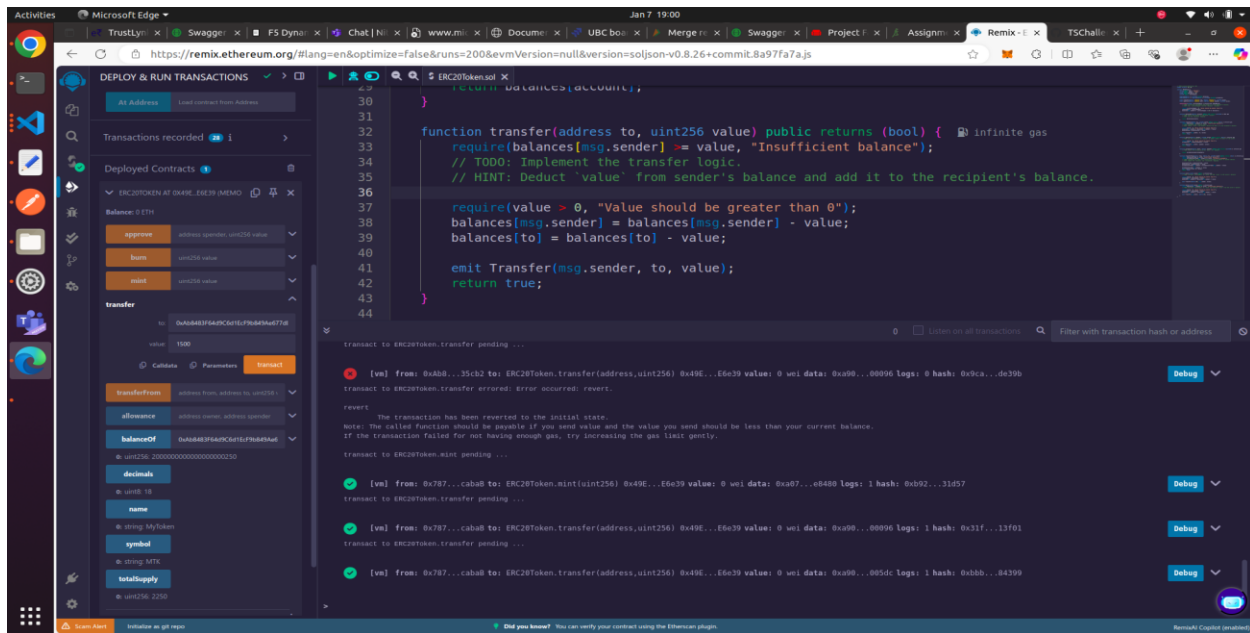
3. Burn: Burnt 250 and now the totalSupply is 2250.



The screenshot shows the Remix IDE interface. On the left, the 'Deployed Contracts' panel lists the 'ERC20Token' contract. The 'Burn' transaction is selected, showing a value of 250. The main editor displays the Solidity code for the 'burn' function. The right sidebar shows the 'Transactions' panel with a list of transactions, including the 'burn' transaction.

```
37 require(value > 0, "Value should be greater than 0");
38 balances[msg.sender] = balances[msg.sender] - value;
39 balances[to] = balances[to] - value;
40
41 emit Transfer(msg.sender, to, value);
42 return true;
43
44
45 function approve(address spender, uint256 value) public returns (bool) { Infinite gas
46 // TODO: Set the allowance for 'spender' to spend 'value' on behalf of the caller.
47
48 require(value > 0, "Value should be greater than 0");
49 allowances[msg.sender][spender] = value;
50
51 emit Approval(msg.sender, spender, value);
```

4. Transfer: Transferring 1500 ethers.



The screenshot shows the Remix IDE interface. On the left, the 'Deployed Contracts' panel lists the 'ERC20Token' contract. The 'Transfer' transaction is selected, showing a value of 1500. The main editor displays the Solidity code for the 'transfer' function. The right sidebar shows the 'Transactions' panel with a list of transactions, including the 'transfer' transaction.

```
30 }
31
32 function transfer(address to, uint256 value) public returns (bool) { Infinite gas
33 require(balances[msg.sender] >= value, "Insufficient balance");
34 // TODO: Implement the transfer logic.
35 // HINT: Deduct 'value' from sender's balance and add it to the recipient's balance.
36
37 require(value > 0, "Value should be greater than 0");
38 balances[msg.sender] = balances[msg.sender] - value;
39 balances[to] = balances[to] + value;
40
41 emit Transfer(msg.sender, to, value);
42 return true;
43
44 }
```

5. Allowance and Approve: Adding allowance for spender upto 20000 and approve.

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is visible, showing the deployed contract 'ERC20Token' and its functions: approve, burn, mint, transfer, and allowance. The 'mint' function is currently selected. The main editor displays the Solidity code for 'ERC20Token.sol'. The code includes functions for minting, burning, approving, and transferring tokens. The transaction logs at the bottom show a failed transaction: '[vm] from: 0x4B8...35c2 to: ERC20Token.transfer(address,uint256) 0x4B8...ab7c value: 0 wei data: 0x00...0012c logs: 0 hash: 0x00b...fe86'. The error message is 'revert: The transaction has been reverted to the initial state. Reason provided by the contract: "Insufficient balance". If the transaction failed for not having enough gas, try increasing the gas limit gently.'

6. Balance: Fetching balance for an account.

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is visible, showing the deployed contract 'ERC20Token' and its functions: approve, burn, mint, transfer, allowance, balanceOf, decimals, name, symbol, and totalSupply. The 'balanceOf' function is currently selected. The main editor displays the Solidity code for 'ERC20Token.sol'. The code includes functions for minting, burning, approving, transferring, and fetching the balance of an account. The transaction logs at the bottom show a failed transaction: '[vm] from: 0x4B2...C03b to: ERC20Token.transferFrom(address,address,uint256) 0x49E...E6c39 value: 0 wei data: 0x23b...0000a logs: 0 hash: 0xddc...0a61c'. The error message is 'revert: The transaction has been reverted to the initial state. Reason provided by the contract: "Allowance exceeded". If the transaction failed for not having enough gas, try increasing the gas limit gently.'

ERROR Cases:

1. Transfer: Insufficient balance

The screenshot displays the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel shows a list of deployed contracts, including ERC20Token, ERC20Token.balanceOf, ERC20Token.transfer, ERC20Token.allowance, and ERC20Token.balanceOf. The 'transfer' function is selected, and its parameters are set to 'address sender, uint256 value'. The 'run' button is highlighted. The main editor shows the Solidity code for the ERC20Token contract, with the transfer function highlighted. The function code is as follows:

```
function transferFrom(address from, address to, uint256 value) public returns (bool) {
    require(balances[from] >= value, "Insufficient balance");
    require(allowances[from][msg.sender] >= value, "Allowance exceeded");
    // TODO: Implement the logic for transferring tokens on behalf of another address.
    // HINT: Update the 'balances' and 'allowances' mappings.

    balances[from] = balances[from] - value;
    balances[to] = balances[to] + value;
    allowances[from][msg.sender] = allowances[from][msg.sender] - value;

    allowances[from][msg.sender] += value;
    emit Transfer(from, to, value);
    return true;
}
```

The transaction logs on the right show two failed transactions. The first transaction, from 0x482...C82db to ERC20Token.transfer(address,uint256) 0x49E...E6e39, failed with the error 'Insufficient balance'. The second transaction, from 0x482...C82db to ERC20Token.transfer(address,uint256) 0x49E...E6e39, also failed with the error 'Insufficient balance'. The logs also show a successful transaction from 0x482...C82db to ERC20Token.balanceOf(address) 0x49E...E6e39, which returned the balance of 0x70a...35c2.

2. TransferFrom: Insufficient Balance

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying the 'transferFrom' function with parameters: from: 0x4820993b481177c7e8f571ccCa8A9a22C02b, to: 0x4820993b481177c7e8f571ccCa8A9a22C02b, value: 10. Below this, the 'allowance' function is also visible. The main editor shows the Solidity code for ERC20Token.sol, with the 'transferFrom' function highlighted. The transaction logs on the right show a successful 'approve' transaction followed by a failed 'transferFrom' transaction. The error message for the failed transaction is 'Insufficient balance'.

```

60 function transferFrom(address from, address to, uint256 value) public returns (bool) {
61     require(balances[from] >= value, "Insufficient balance");
62     require(allowances[from][msg.sender] >= value, "Allowance exceeded");
63     // TODO: Implement the logic for transferring tokens on behalf of another address.
64     // HINT: Update the 'balances' and 'allowances' mappings.
65
66     balances[from] = balances[from] - value;
67     balances[to] = balances[to] + value;
68     allowances[from][msg.sender] = allowances[from][msg.sender] - value;
69
70     allowances[from][msg.sender] -= value;
71     emit Transfer(from, to, value);
72     return true;
73 }
74

```

Transaction logs:

- [v] from: 0x787...cab8 to: ERC20Token.approve(address,uint256) 0x49c...E639 value: 0 wei data: 0x095...04e2 logs: 1 hash: 0xb68...a21a4
- [v] from: 0x4820993b481177c7e8f571ccCa8A9a22C02b to: ERC20Token.allowance(address,address) data: 0x0d6...35cb2
- [v] from: 0x4820993b481177c7e8f571ccCa8A9a22C02b to: ERC20Token.allowance(address,address) data: 0x0d6...35cb2
- [v] from: 0x482...C02b to: ERC20Token.transferFrom(address,address,uint256) 0x49c...E639 value: 0 wei data: 0x23b...0000a logs: 0 hash: 0x4b...d79d9
- transaction to ERC20Token.transferFrom error: Error occurred: revert.
- Reason provided by the contract: "Insufficient balance"
- If the transaction failed for not having enough gas, try increasing the gas limit gently.

3. Allowance: Allowance exceeded.

The screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel is active, displaying the 'transferFrom' function with parameters: from: 0x4820993b481177c7e8f571ccCa8A9a22C02b, to: 0x4820993b481177c7e8f571ccCa8A9a22C02b, value: 10. Below this, the 'allowance' function is also visible. The main editor shows the Solidity code for ERC20Token.sol, with the 'transferFrom' function highlighted. The transaction logs on the right show a successful 'approve' transaction followed by a failed 'transferFrom' transaction. The error message for the failed transaction is 'Allowance exceeded'.

```

60 function transferFrom(address from, address to, uint256 value) public returns (bool) {
61     require(balances[from] >= value, "Insufficient balance");
62     require(allowances[from][msg.sender] >= value, "Allowance exceeded");
63     // TODO: Implement the logic for transferring tokens on behalf of another address.
64     // HINT: Update the 'balances' and 'allowances' mappings.
65
66     balances[from] = balances[from] - value;
67     balances[to] = balances[to] + value;
68     allowances[from][msg.sender] = allowances[from][msg.sender] - value;
69
70     allowances[from][msg.sender] -= value;
71     emit Transfer(from, to, value);
72     return true;
73 }
74

```

Transaction logs:

- [v] from: 0x482...C02b to: ERC20Token.mint(uint256) 0x49c...E639 value: 0 wei data: 0xa07...9c308 logs: 1 hash: 0x703...a6119
- [v] from: 0x482...C02b to: ERC20Token.transferFrom(address,address,uint256) 0x49c...E639 value: 0 wei data: 0x23b...0000a logs: 0 hash: 0x181...b79d6
- transaction to ERC20Token.transferFrom error: Error occurred: revert.
- Reason provided by the contract: "Allowance exceeded"
- If the transaction failed for not having enough gas, try increasing the gas limit gently.
- transaction to ERC20Token.transferFrom pending ...
- [v] from: 0x482...C02b to: ERC20Token.transferFrom(address,address,uint256) 0x49c...E639 value: 0 wei data: 0x23b...0000a logs: 0 hash: 0x4dc...0a61c
- transaction to ERC20Token.transferFrom error: Error occurred: revert.
- Reason provided by the contract: "Allowance exceeded"
- If the transaction failed for not having enough gas, try increasing the gas limit gently.

