

**FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING**  
**Department of Computer Engineering**

**1. Course , Subject & Experiment Details**

<b>Academic Year</b>	<b>2022-23</b>	<b>Estimated Time</b>	<b>02 - Hours</b>
<b>Course &amp; Semester</b>	<b>B.E. (CMPN)- Sem VII</b>	<b>Subject Name &amp; Code</b>	<b>BCT - (CSDC7022)</b>
<b>Chapter No.</b>	<b>03</b>	<b>Chapter Title</b>	<b>Programming for Blockchain</b>

<b>Practical No:</b>	<b>4</b>
<b>Title:</b>	Auction for fundraising using Solidity
<b>Date of Performance:</b>	<b>29/08/2022</b>
<b>Date of Submission:</b>	<b>05/09/2022</b>
<b>Roll No:</b>	<b>8953</b>
<b>Name of the Student:</b>	<b>Brendan Lucas</b>

**Evaluation:**

<b>Sr. No</b>	<b>Rubric</b>	<b>Grade</b>
<b>1</b>	<b>On time submission Or completion (2)</b>	
<b>2</b>	<b>Preparedness(2)</b>	
<b>3</b>	<b>Skill (4)</b>	
<b>4</b>	<b>Output (2)</b>	

**Signature of the Teacher:**

**Date:**

## Code:

```
// SPDX-License-Identifier: GPL-3.0
```

```
pragma solidity ^0.8.7;
```

```
contract AUC{
    mapping(uint => uint) public token_map;
    mapping(uint => mapping(uint => uint)) public item_token;
    mapping(uint => uint) public item_owner;
    uint public nun;
    constructor () {
        token_map[0]=5;
        token_map[1]=5;
        token_map[2]=5;
        token_map[3]=5;
        token_map[4]=5;

        item_token[0][0] = 0;
        item_token[1][0] = 0;
        item_token[2][0] = 0;
        item_token[3][0] = 0;
        item_token[4][0] = 0;
        item_token[0][1] = 0;
        item_token[1][1] = 0;
        item_token[2][1] = 0;
        item_token[3][1] = 0;
        item_token[4][1] = 0;
        item_token[0][2] = 0;
        item_token[1][2] = 0;
        item_token[2][2] = 0;
        item_token[3][2] = 0;
        item_token[4][2] = 0;

        item_owner[0]=0;
        item_owner[1]=0;
        item_owner[2]=0;

    }

    function auction(uint _bidder,uint _bid,uint _item) public {
        if(token_map[_bidder]>=_bid)
```

```

    {
        token_map[_bidder] -= _bid;
        item_token[_bidder][_item] += _bid;
    }
    else{
        revert("Not Sufficient Tokens");
    }
}

function select_winner(uint _item) public {
    nun = block.timestamp%4+1;
    if(item_token[nun][_item]>0)
    {
        item_owner[_item] = nun;
    }
    else{
        revert();
    }
}
}

```

## Output

```

creation of AUC pending...

[✓] [vm] from: 0x488...35c82 to: AUC.(constructor) value: 0 wei data: 0x088...70033 logs: 0 hash: 0xf3...612c4
transaction to AUC.auction pending ...

[✓] [vm] from: 0x787...cab88 to: AUC.auction(uint256,uint256,uint256) 0x899...5a595 value: 0 wei data: 0xb73...00083 logs: 0 hash: 0x2f5...dff24
transaction to AUC.auction pending ...

[✓] [vm] from: 0x787...cab88 to: AUC.auction(uint256,uint256,uint256) 0x899...5a595 value: 0 wei data: 0xb73...00084 logs: 0 hash: 0xae2...8dfc
transaction to AUC.auction pending ...

[✓] [vm] from: 0x787...cab88 to: AUC.auction(uint256,uint256,uint256) 0x899...5a595 value: 0 wei data: 0xb73...00081 logs: 0 hash: 0x524...ebfd
transaction to AUC.auction pending ...

[✗] [vm] from: 0x787...cab88 to: AUC.auction(uint256,uint256,uint256) 0x899...5a595 value: 0 wei data: 0xb73...00080 logs: 0 hash: 0x955...7bdf1
transaction to AUC.auction errored: VM error: revert.

revert
  The transaction has been reverted to the initial state.
  Reason provided by the contract: "Not Sufficient Tokens".
  Debug the transaction to get more information.

call to AUC.item_token

0x1 [call] from: 0x7871103ca0b7e34ac0f82642a7cc18a495cab88 to: AUC.item_token(uint256,uint256) data: 0x505...00082
call to AUC.item_token

0x2 [call] from: 0x7871103ca0b7e34ac0f82642a7cc18a495cab88 to: AUC.item_token(uint256,uint256) data: 0x505...00081
call to AUC.item_token

0x3 [call] from: 0x7871103ca0b7e34ac0f82642a7cc18a495cab88 to: AUC.item_token(uint256,uint256) data: 0x505...00083
transaction to AUC.select_winner pending ...

[✓] [vm] from: 0x787...cab88 to: AUC.select_winner(uint256) 0x099...5a595 value: 0 wei data: 0x022...00003 logs: 0 hash: 0xc12...9a806
call to AUC.run

0x4 [call] from: 0x7871103ca0b7e34ac0f82642a7cc18a495cab88 to: AUC.run() data: 0x087...b49c0

```