

AIM:

Implement a smart contract for Auction Application which have constructor that initialize min bid to 30 ethers. Also implement place bid function which will allow the customer to place bid. Also implement cancel auction which will cancel the auction. Implement withdraw and complete auction function. The withdraw function will be called once auction will be cancelled by the owner. The complete function will display the highest bid at the time of calling. Compile and Test the smart contract with different test cases and show the output. Admit at least 3 customers and show the output.

PROGRAM CODE:

```
pragma solidity ^0.7.6;

contract Auction {
    uint[] public prices = new uint[](10);
    uint public count = 0;
    uint public winner;
    uint public price;
    string public cancelled;
    string public _status;

    constructor() {
        cancelled = "";
        prices[0] = 30;
    }

    function cancelAuction () public {
        withdrawAuction();
        count = 0;
        cancelled = "Auction is cancelled";
    }
}
```

```
}
```

```
function withdrawAuction() public {  
    prices = new uint[](10);  
    prices[0] = 30;  
}
```

```
function completeAuction() public {  
    uint maxi = 0;  
    uint win;  
    for(uint x = 0; x < prices.length; x++)  
    {  
        if(prices[x] > maxi)  
        {  
            win = x;  
            maxi = prices[x];  
        }  
    }  
    withdrawAuction();  
    count = 0;  
    winner = win;  
    price = maxi;  
}
```

```
function placeBid(uint _price) public {  
    if(_price > 30)  
    {  
        prices[count] = _price;
```

```

        _status = "Successful";
    }
    else
    {
        _status = "At least 30 price is required";
    }
    count++;
}
}

```

OUTPUT:

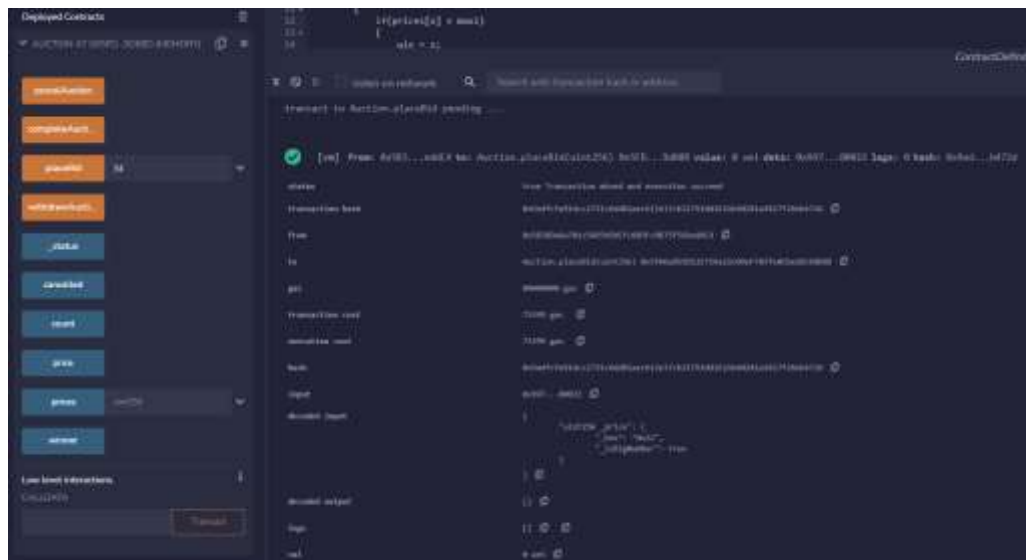


Figure 1: Place Bid 1

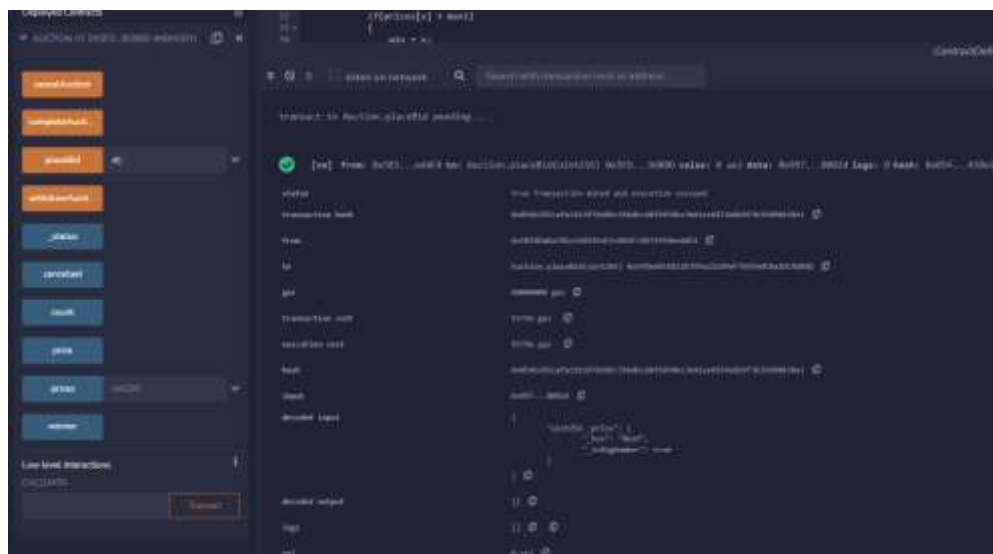


Figure 2: Place Bid 2

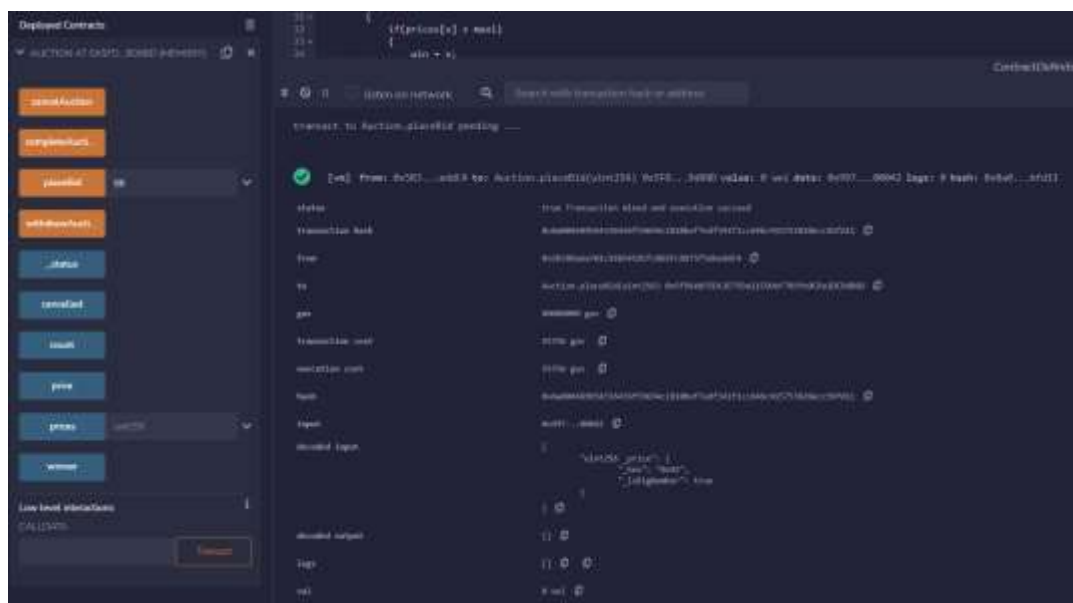


Figure 3: Place Bid 3

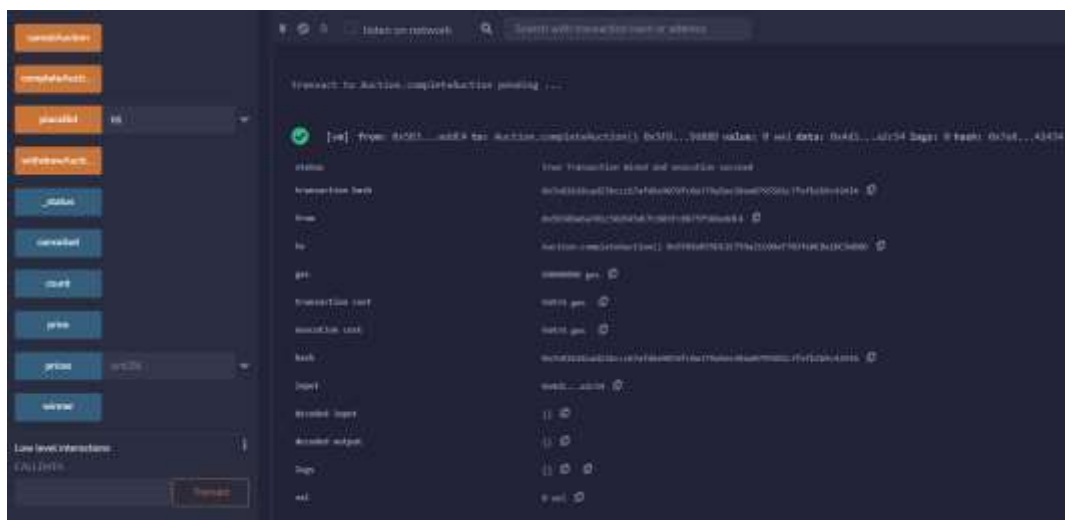


Figure 4: Completing Auction

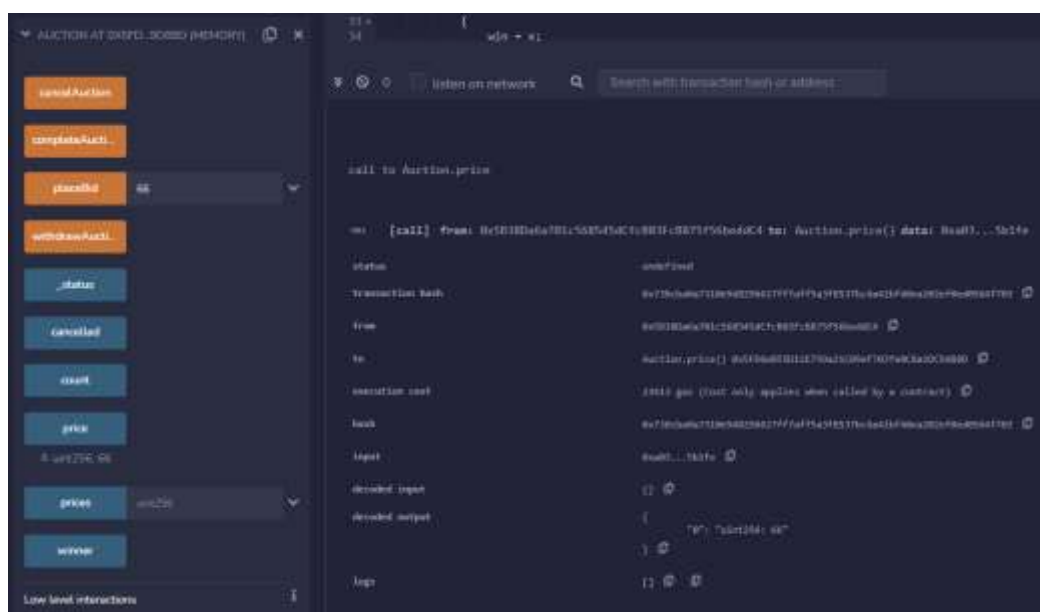


Figure 5: Final Output