

Solidity 13 November Assignment

**Q- Remove array element by shifting elements from right to left**

**remove(n) in the comments below indicates remove elements from the nth index of the array**

**examples of suggested inputs and outputs-**

***// [1, 2, 3] -- remove(1) --> [1, 3, 3] --> [1, 3]***

***// [1, 2, 3, 4, 5, 6] -- remove(2) --> [1, 2, 4, 5, 6, 6] --> [1, 2, 4, 5, 6]***

***// [1, 2, 3, 4, 5, 6] -- remove(0) --> [2, 3, 4, 5, 6, 6] --> [2, 3, 4, 5, 6]***

***// [1] -- remove(0) --> [1] --> []***

***—---------------Answer —------------------------***

**//SPDX-License-Identifier: MIT**

**pragma solidity ^0.8.3;**

**contract ArrayShift{**

**uint[] public arr;**

**function example() public{**

**arr = [1,2,3];**

**delete arr[1]; // [1,0,3]**

**}**

**// [1,2,3] --- remove(1) ---> [1,3,3] --> [1,3]**

***// [1, 2, 3, 4, 5, 6] -- remove(2) --> [1, 2, 4, 5, 6, 6] --> [1, 2, 4, 5, 6]***

***// [1, 2, 3, 4, 5, 6] -- remove(0) --> [2, 3, 4, 5, 6, 6] --> [2, 3, 4, 5, 6]***

***// [1] -- remove(0) --> [1] --> []***

**function remove(uint \_index) public{**

**require (\_index < arr.length, "index out of bound");**

**for (uint i = \_index; i < arr.length - 1; i++){**

**arr[i] = arr[i+1];**

**}**

**arr.pop();**

**}**

**function test() external{**

**arr = [1,2,3,4,5];**

**remove(2); // [1,2,4,5]**

**assert (arr[0] == 1);**

**assert (arr[1] == 2);**

**assert (arr[2] == 4);**

**assert (arr[3] == 5);**

**assert (arr.length == 4);**

**arr = [1];**

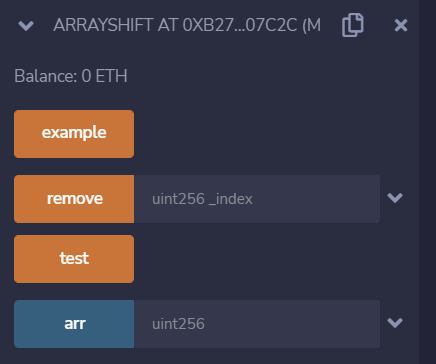
**remove(0); //[]**

**assert(arr.length == 0);**

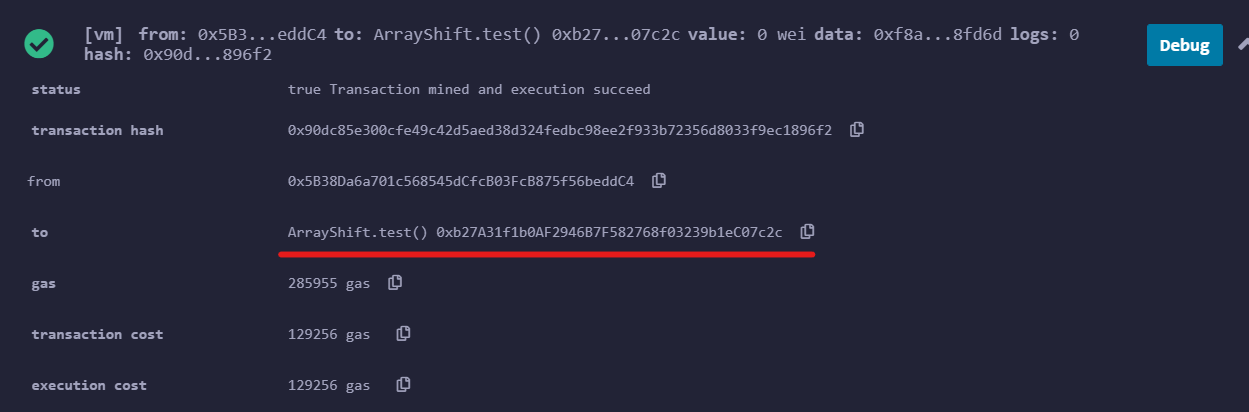
**}**

**}**

**Deployed contract**

****

**test succeeded**

****