



```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 contract FallbackExample {
5     uint256 public result;
6
7     // Fallback function must be declared as external.
8     fallback() external payable {
9         result = 1;
10    }
11
12    receive() external payable {
13        result = 2;
14    }
15 }
16
17 // Explainer from: https://solidity-by-example.org/fallback/
18 // Ether is sent to contract
19 // is msg.data empty?
20 //      / \
21 //    yes no
22 //      / \
23 // receive()? fallback()
```

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.6.0;
3
4 contract SafeMathTester{
5     uint8 public bigNumber = 255; // unchecked
6
7     function add() public {
8         bigNumber = bigNumber + 1;
9     }
10 }
```

0 ☐ listen on all transactions Search with transaction hash or address

[vm] from: 0x583...eddC4 to: PriceConverter.(constructor) value: 0 wei data: 0x605...e033 logs: 0 hash: 0x768...8b061 Debug

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract SafeMathTester{
5     uint8 public bigNumber = 255; // checked
6
7     function add() public {
8         unchecked {bigNumber = bigNumber + 1;}
9     }
10 }
```

0 ☐ listen on all transactions Search with transaction hash or address

[vm] from: 0x583...eddC4 to: PriceConverter.(constructor) value: 0 wei data: 0x605...e033 logs: 0 hash: 0x768...8b061 Debug