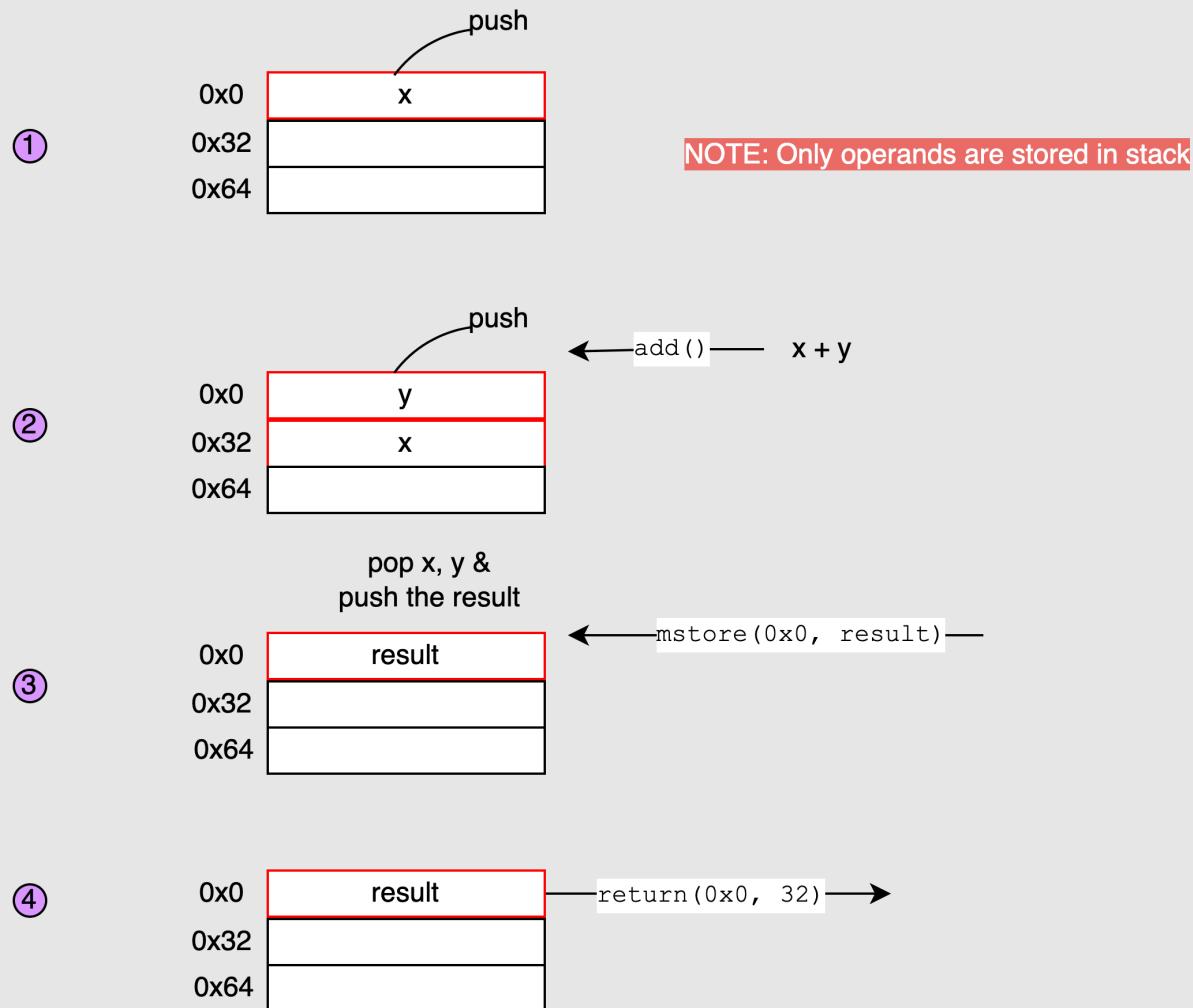


Understanding ADD operation



```
● ● ●  
contract Arithmetic {  
    /// @dev Simple Add operation  
    function add(uint256 x, uint256 y) public pure returns (uint256 z)  
    {  
        assembly {  
            z := add(x, y)  
        }  
    }  
}
```

```
● > forge test --match-test testAdd  
[#] Compiling...  
[#] Compiling 2 files with 0.8.18  
[#] Solc 0.8.18 finished in 931.40ms  
Compiler run successful!  
  
Running 1 test for test/Yul1.t.sol:Yul1Test  
[PASS] testAdd() (gas: 5506)  
Test result: ok. 1 passed; 0 failed; 0 skipped; finished in 2.73ms  
Ran 1 test suites: 1 tests passed, 0 failed, 0 skipped (1 total tests)
```

```
● ● ●  
contract Arithmetic {  
    /// @dev Simple Add operation  
    function add(uint256 x, uint256 y) public pure returns (uint256)  
    {  
        assembly {  
            let result := add(x, y)  
            mstore(0x0, result)  
            return(0x0, 32)  
        }  
    }  
}
```

```
● > forge test --match-test testAdd  
[#] Compiling...  
[#] Compiling 2 files with 0.8.18  
[#] Solc 0.8.18 finished in 925.63ms  
Compiler run successful!  
  
Running 1 test for test/Yul1.t.sol:Yul1Test  
[PASS] testAdd() (gas: 5485)  
Test result: ok. 1 passed; 0 failed; 0 skipped; finished in 1.11ms  
Ran 1 test suites: 1 tests passed, 0 failed, 0 skipped (1 total tests)
```