

# Homework 6 Answers

## Answer









1.

```
// SPDX-License-Identifier: UNLICENSED language-js
pragma solidity ^0.8.0;

contract YulEx1 {

    function getETHValue() public payable returns(uint value)
    {
        assembly {
            value := callvalue()
        }
    }
}
```

2. The contract simulates cell division, creating 2 child contracts then self destructing. More details are [here](#)

```
1  # ■ => ■ + ■
2  # Minimal runtime bytecode for a contract that mutates
3  # into two child contracts and then self-destructs
4  # 1st child contract receives the call value
5  # 2nd child contract receives the remaining balance
6  # author: Saw-mon and Natalie
7
8  # constructor payload for the spawned contract
9
10 #  push1 RUNTIME_BYTECODE_LEN      # L
11 #  dup1                          # L L
12 #  push1 RUNTIME_BYTECODE_OFFSET  # 0 L L
13 #  returndatasize                 # 0 0 L L
14 #  codecopy                       # L
15 #  returndatasize                 # 0 L
16 #  return
17 push9 0x601e8060093d393df3
18 #  LL----- The only important byte that varies.
```

Note CREATE opcode uses these stack values

1. `value` : value in [wei](#) to send to the new account.
2. `offset` : byte offset in the [memory](#) in bytes, the initialisation code for the new account.
3. `size` : byte size to copy (size of the initialisation code).

Question 3 :

Please refer to the following link:

<https://docs.soliditylang.org/en/v0.8.17/yul.html#complete-erc20-example>

The function in question is used to calculate the storage location (offset) for both the account that owns the ERC20 and the allowed account(spender). The maximum amount that is allowed to be spent is stored at this offset.

The function is used in another as follows:

```
function setAllowance(account, spender, amount) {  
  sstore(allowanceStorageOffset(account, spender), amount)  
}
```

The `allowanceStorageOffset` function is called to work out the location in storage using `sstore` with the corresponding amount in the `setAllowance` function.