Answers to homework 6

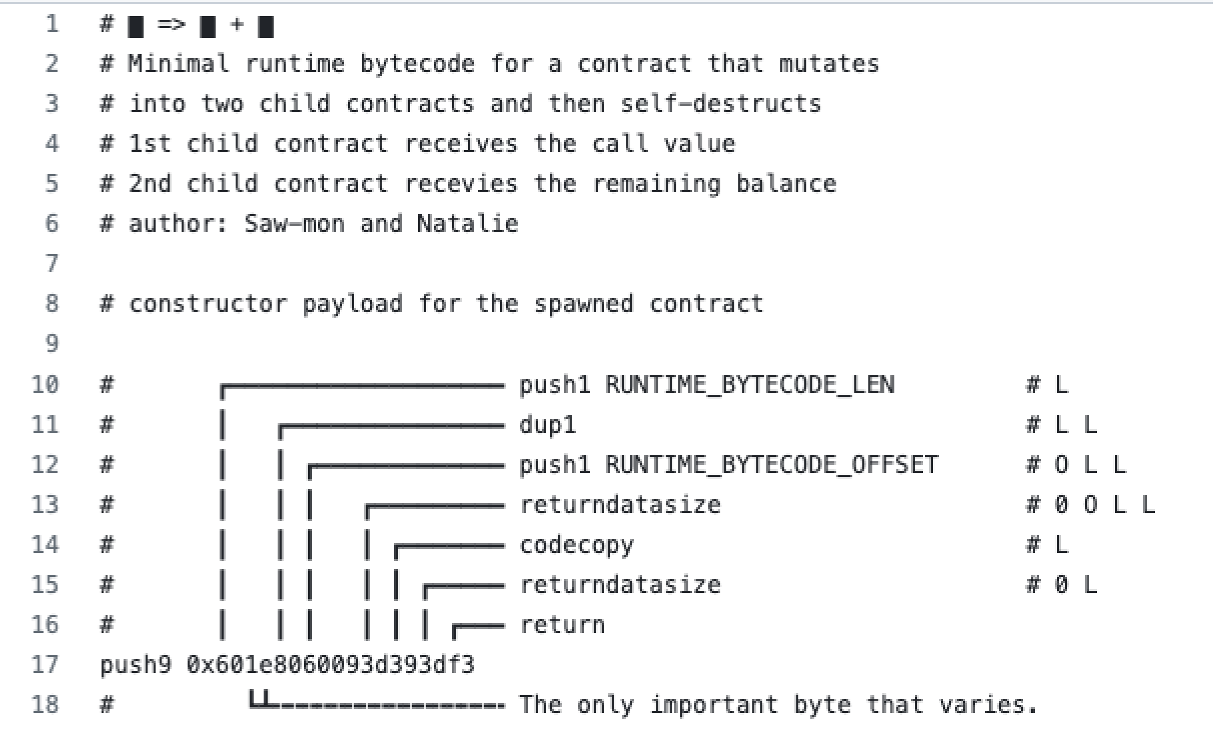
1.

```
// SPDX-License-Identifier: UNLICENSED
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



```
# █ => █ + █
# Minimal runtime bytecode for a contract that mutates
# into two child contracts and then self-destructs
# 1st child contract receives the call value
# 2nd child contract recevies the remaining balance
# author: Saw-mon and Natalie

# constructor payload for the spawned contract

#       ┌─────────────────── push1 RUNTIME_BYTECODE_LEN         # L
#       │   ┌─────────────── dup1                               # L L
#       │   │ ┌───────────── push1 RUNTIME_BYTECODE_OFFSET      # O L L
#       │   │ │   ┌───────── returndatasize                     # 0 O L L
#       │   │ │   │ ┌─────── codecopy                           # L
#       │   │ │   │ │ ┌───── returndatasize                     # 0 L
#       │   │ │   │ │ │ ┌─── return
push9 0x601e8060093d393df3
#         └┴---------------- 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.