

**Q1 In the `safeTransferFrom` function, what does `0x23b872dd000000000000000000000000` represent and what does it mean when used in the following context on line 192: `mstore(0x0c, 0x23b872dd000000000000000000000000) .?`**

`0x23b872dd000000000000000000000000` funcsig `transferFrom(address,address,uint256)` store value `0x23b872dd000000000000000000000000` at location `0x0c` (12)

**Q2 In the `safeTransferFrom` function, why is `shl` used on line 191 to shift the `from` to the left by 96 bits?**

move `from` address value to `0x2c` next to function signature at `0x1c` and eliminate dirty bytes.

**Q3 In the `safeTransferFrom` function, is this memory safe assembly? Why or why not?**

although it can be considered safe as `0x60` was temporarily used but reset it deviates from best practices

**Q4 In the `safeTransferFrom` function, on line 197, why is `0x1c` provided as the 4th argument to `call` ?**

it contains start of function signature ( `transferFrom(address,address,uint256)` ). from `0x1c` next 100 bytes contains abi-encoded low level call.

**Q5 In the `safeTransfer` function, on line 266, why is `revert` used with `0x1c` and `0x04` ?**

starting at `0x1c` onwards 4 bytes are `90b8ec18` which refers to `transferFailed()` error which is given as revert reason.

**Q6 In the `safeTransfer` function, on line 268, why is `0` `mstore'd` at `0x34` .?**

when `mstore(0x34, amount)` is called it also writes at regions `0x40` to `0x54` which belongs to the free memory pointer( `0x40` ). `mstore(0x34, 0)` is used to overwrite that region back with zeroes.

**Q7 In the `safeApprove` function, on line 317, why is `mload(0x00)` validated for equality to 1?**

it is checking for successful return after calling approve function.

**Q8 In the `safeApprove` function, if the `token` returns `false` from the `approve(address,uint256)` function, what happens?**

It will revert with `ApproveFailed()` Error.