University of Edinburgh	*Fall 2021-22*
Blockchains & Distributed Ledgers	

Assignment #1 (Total points = 100)

Due: Monday 11.10.2021, 16.30

Part 1: Theory (5 x 12 points)

1. How are hash functions used in a Merkle Tree?
    a. Describe (or sketch) how a (binary complete) Merkle Tree is constructed for the following 5 chunks of data: ABC, DEF, GHI, KLM, OPQ
    b. Describe (or sketch) how a Patricia Trie is constructed for the following key/value store: {bla: 17, blabla: 28, bored: 53, board: 39, aboard: 42, abroad: 17}
2. Derive the formula for the birthday paradox (show your work, explaining every step) and calculate the number of elements needed to find a collision with at least 50%. Apply this to find out how many Bitcoin users are needed to initialize their wallet's seed, which is based on a random selection of 12 random words from the list in https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt, to have the event that, with probability at least 50%, at least two users end up with exactly the same seed.
3. A miner creates a block B which contains address α, on which he wants to receive his rewards. An attacker changes the contents of B, such that instead of α it defines a new address α', which is controlled by the attacker. Will the attacker receive the rewards that the miner tries to claim and why (or why not)? Give a detailed explanation of your answer.
4. Give a detailed example of how an orphan block can be created in Bitcoin.
5. A signature scheme is EU-CMA secure if, *for every probabilistic polynomially-bounded adversary A*, A cannot win the EU-CMA security game. Write a brief description of the EU-CMA game. Then, consider the following example:
   Assume a digital signature scheme Σ = *(KeyGen, Sign, Verify)* with security parameter λ (256 bits). Alice generates a pair of verification/signing keys *(vk, sk)* via KeyGen and sends *vk* to Mallory and Bob. Then, consider the following scenarios:
    a. Mallory sends Alice a message *m* and requests a signature. Alice runs *Sign(m, sk)* responds with *(m, σ, vk)*. Then, Mallory sends Bob *(m', σ', vk)*, where $m' \neq m$ and $\sigma' \neq \sigma$, and Bob accepts it as validly signed.
    b. Mallory sends Alice two messages *m, m'*, where $m' \neq m$, and requests a signature for each. Alice runs *Sign(m, sk)* and *Sign(m', sk)* and responds with *(m, σ, vk)* and *(m', σ', vk)* respectively. Then, Mallory sends Bob *(m', σ', vk)* and Bob accepts it as validly signed.

   For each case, explain in detail whether, based only on the described scenario, Mallory wins the EU-CMA game. In both cases (if yes or if no), can we conclude whether Σ satisfies the EU-CMA property? If not, why not? If yes, how is this proven, given only the described scenario? If we cannot conclude whether it is secure, explain what information is missing.

Part 2: Hands-on (10 + 30 points)

1. Using the course's private Ethereum chain, send 1 ETH to the address of a fellow student. Write a small description on how you conducted the payment, including the transaction's id and addresses which you used. (Instructions on how to connect to the course's private chain are available here)

2. A smart contract has been deployed on the course's private chain. You may find its code here and its deployed address is: *0xA8D9D864dA941DdB53cAed4CeB8C8Bcf53aFe580* You can compile and interact with it using Remix. You should successfully create a transaction that interacts with the contract, either depositing to or withdrawing from it some coins. Describe briefly the contract's functionality (that is, the purpose of each variable, function, and event) and provide the id of the transaction you performed and the address you used.