University of Edinburgh	*Fall 2022-23*
Blockchains & Distributed Ledgers	

Assignment #1 (Total points = 100)

Due: Monday 17.10.2022, 12.00 (noon)

Please remember the good scholarly practice requirements of the University regarding work for credit. You can find guidance at the School page: https://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

Part 1: Theory (5 x 14 points)

1.
    a. Formally prove (or disprove) the following proposition: “It is infeasible to forge a proof of inclusion for a Merkle Tree that uses a collision resistant hash function.”
    b. Describe (or sketch) i) a binary full Merkle Tree, ii) a binary complete Merkle tree, for the following chunks of data: *A, B, C, D, E*. Provide the proof of inclusion for *E* in both cases.
2. Derive the formula for the birthday paradox (show your work, explaining every step) and calculate the approximate number of elements needed to find a collision with at least 50% probability. Apply this to find out approximately how many Bitcoin wallets needed to initialize their seed, which is a randomly sampled sequence of 12 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 wallets end up with exactly the same seed (calculate both with and without replacement of sampled words).
3. Assume that all Bitcoin miners are honest except one (the attacker). A miner creates a block B which contains address α, on which they want to receive their rewards. The attacker changes the contents of B, replacing α with an adversarial address α’. What needs to happen for the attacker to receive the rewards for B?
4. Give two detailed examples of how an orphan block can be created in Bitcoin.
5. Construct a digital signature scheme, based on a hash function, that is one-time secure (i.e., it is secure if each private key signs only a single message). Describe in detail how the keys are generated and how signatures are issued and verified.

Part 2: Hands-on (5 + 25 points)

(Instructions on how to connect to the course’s testnet are available here.)

6. Using the course’s Ethereum testnet, 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.
7. A smart contract has been deployed on the course’s testnet. Its code is available 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 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.