CLASS TEST

DECENTRALISED TECHNOLOGIES AND CRYPTOCURRENCIES

MAY 23RD 2022

LAB TEST INSTRUCTIONS:

* *Test can be taken in English or Italian.*
* *Fill in below with your name, surname, and student number*
* *Answer concisely to questions. Do not use fonts smaller that 11 pt. Do not exceed 4 pages for your answers.*
* *This test lasts 60 min.*
* *Please, save your word file regularly.*
* *The exam test will be collected at the end of the test.*

Name and Surname: TEO BUCCI

Student Number: 10621873

**Questions**

1. Give a definition of decentralised ledger. Explain how it is used in blockchain protocols and why it is different from a standard database.

1. List and explain three (3) properties of the distributed ledger, making an example of application for each of them.

1. Explain what are Elliptic Curves and how they are used in Bitcoin.

1. Explain the following concepts: smart contract, Dapp, and wallet. Then describe a use case hat uses at least two of them.

1. In Remix, modify the code below so that the sender function will be able to transfer tokens to three accounts at the same time instead of one, splitting the amount into three. Please implement required changes, compile, deploy and test it with Remix. Report a screenshot of the new function. Explain also your modification to the code.

**CODE**

**pragma solidity ^0.5.0;**

**contract Coin {**

**// The keyword "public" makes those variables // easily readable from outside. address public minter;**

**mapping (address => uint) public balances;**

**// Events allow light clients to react to // changes efficiently.**

**event Sent(address from, address to, uint amount);**

**// This is the constructor whose code is // run only when the contract is created.**

**constructor() public {**

**minter = msg.sender;**

**}**

**function mint(address receiver, uint amount) public { require(msg.sender == minter); require(amount < 1e60);**

**balances[receiver] += amount;**

**}**

**function send(address receiver, uint amount) public {**

**require(amount <= balances[msg.sender], "Insufficient balance."); balances[msg.sender] -= amount; balances[receiver] += amount;**

**emit Sent(msg.sender, receiver, amount);**

**}**

**}**

**ASNWERS**

1….

2…..

….