**Problem 1:**
Let’s build a banking system smart contract using the solidity concepts we learnt. Your bank will provide facilities that a regular commercial bank provides like taking deposits from customers and giving them a certain rate of interest, allowing them to deposit, withdraw and transfer money (Ethereum here).
Your bank should have the following functions and specifications:

- A function called **ShowTotalBalaInce** should return the total Ethereum (a uint) stored in the bank at that point in time.
- A function called **ShowBalance** should return the balance of a user when *address* of a user is given as input.
- A function called **AddBalance** to add funds to an account.
- A transfer function called **Transfer** transfers a given amount of Ethereum funds from *msg.sender* to a given *address*.
- A function called **Withdraw** to withdraw Ethereum and put it back to *msg.sender* ‘s account.
- As when you deposit money in the bank you get a certain ROI, similarly our bank will also give 7% simple interest on deposit, so whenever a user calls the **GetBalance** function make sure you return the value after the interest is added.
  Hint:- To implement this you will require the knowledge of working with timestamps
  Watch this https://youtu.be/w5yjHemMS5g video.
- Create events for all the functions you create

**Problem 2:**
We all are super senti about the election that takes place at IIT-Bombay for various important positions, so certainly it’s super important to make sure voting happens in a fair fashion and transparency. Smart contracts come to the rescue. We can build a robust voting system using solidity with very few lines of code relatively. So write a smart contract that can be used in the election with all the obvious and necessary functionalities.
It’s a bit open-ended question, try to use whatever you have learned to implement your ideas.