This is an **individual assessment**. Do not discuss with other students. Any form of academic dishonesty, including plagiarism, collaborating, copying and even sharing with others, will be considered a major offense and will result in a F grade for the module.

Problem: We want to build a system such that people can pool their money together to invest. One individual will be given the right to invest, and this individual is generated from majority voting. Each individual's voting power is proportional to the money he or she puts in the pool. Specifically, to implement this system, we want to have a Pool contract such that

1) It issues a ERC20 token PT (PoolToken) with the price of 0.01 Ether each. There is no limit of supply to PT (anyone can get PT at the price of 0.01 Eth at any time). Assume the amount of PT is always an integer.
2) An address can send money to Pool in the form of PT. An address can send PTs to the Pool multiple times.
3) Each PT grants the sender a vote.
4) Each address can propose one address to vote for. All the votes from the former address will be counted for the later address. (An address, even with multiple votes, cannot vote for multiple candidates.)
5) The chairperson of Pool is the deployer of Pool contract, set at the time of deployment
6) Only the chairperson can call the end of voting.
7) When the voting ends, count the votes. If there is a candidate gets strictly over 50% of the votes received (not amount of PT received, since someone may not vote), this candidate is the winner. Transfer all the PTs in the Pool to the winner.
8) If no candidate gets strictly over 50% of the votes received, the voting is drawn. Return PT to their previous owners.

Please

1) work on Pool.sol to implement the system above. (10pts) Hint:
    a. ERC20.sol need not to be edited;
    b. Please look at the comments in the skeleton code, which briefly provides the logic

c. The function headers need not to be edited.

d. You can check the test file to get more specific ideas about the requirements

e. Please throw the exact error messages when supplied in the skeleton comments

2) work on test_pool.js on two additional tests, ‘Test VoteWon’ and ‘Test VoteDrawn’. You don’t need to edit the other tests.

   a. For ‘Test VoteWon’, test that “VoteWon” is emitted and that all the PTs in the pool are transferred to the winner. Return “VoteWon not working” if PTs are not transferred correctly. Continue to use the case in the other tests. (2pts)

   b. For ‘Test VoteDrawn’, test that “VoteDrawn” is emitted and that the PTs are returned to their previous owners. Return “VoteDrawn not working” if PTs are not returned correctly. Propose a new case where there are three voters. Each of them gets 100 PTs and sends 100PTs to Pool. Each of them votes for a different address, which results in a draw. (3pts)

Submission: Please submit a zip folder with the folder name of [name_student number]. The zip folder should contain only two files, Pool.sol and test_pool.js. Do not change the file name of these two files. Also be reminded to submit the screen recordings.