Challenge 1

The first challenge is based around depositing secret messages from authorised users.

The requirements of the challenge are to write a contract as specified below, plus tests to show the code is working.

The process is as follows :

1. An administrator will add eligible addresses via a UI.
   There will be a maximum of 100 eligible addresses.
   You need to write a function in a contract to store the addresses, the function will have one parameter which is the address.
   Eligible addresses should be stored in a suitable data structure.
2. A user with an eligible address can deposit a secret message of a certain format.
   The message contains 6 flags at the end, each of size 1 bit. The rest of the message can be any number.
   You need to write a function to store the messages, the function will have one parameter which is the message.
   The message should be a Field, we use the last 6 bits as flags.
3. The flags should be checked according to the following rules :
   If flag 1 is true, then all other flags must be false

   If flag 2 is true, then flag 3 must also be true.
   If flag 4 is true, then flags 5 and 6 must be false.

4. You should check that
   Addresses that are not eligible cannot deposit a message
   An address can only deposit one message.

If the above rules are passed, then an event should be emitted to show that a message has been received, and a counter updated to store the number of messages received. The sender's address and the message should be stored in a suitable data structure.

Deliverables

A smart contract with the following functions
i) A function to store eligible addresses
ii) A function to check and store messages
iii) Tests to test these functions

Submitting your solution

You should send us a link to the github repo containing the solution.

Deadline

Your solution needs to be submitted by midnight (UTC) on 31st January 2024.

If you have any questions, please ask on Discord.