Get acquainted with the data types in Solidity

Task description:

Create a smart contract with a majority of different data types in it.

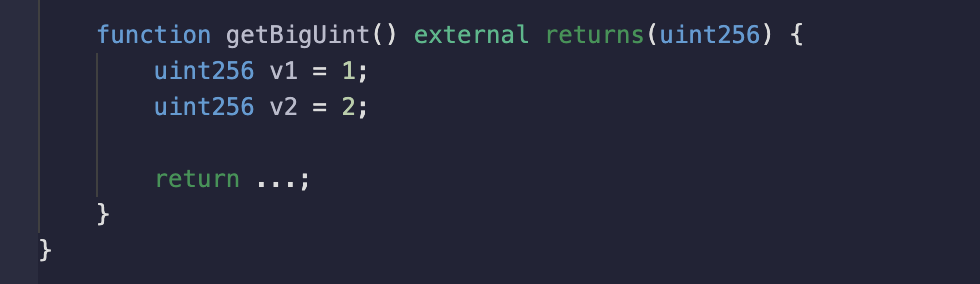
1. Each type should be declared as a storage variable - meaning not inside the function but inside the contract itself.
2. Also, SC should have functions to return all declared values. Interface for those functions will be provided and your contract should inherit it.
3. The interface will be provided in the task folder as a .sol file.
4. Each variable should have **non-zero** value!
5. The string field should contain the words - “Hello World!”.

The smart contract should also have one interesting function. **Read the instructions carefully**.

1. Function getBigUint() should return the uint256 value bigger than 1\_000\_000.   
     
   For doing this you will only have 2 fields inside the function:  
     
   uint256 v1 = 1

uint256 v2 = 2

1. Any arithmetical operator can be used only once, as an example:  
     
   return v1 + v2 - ok  
   return v1 + v2 + v2 - not ok.
2. Only values v1 and v2 can be used.

  
  
Once you are sure your contract compiles and works correctly, your next step is to deploy it on a test network. Will use Sepolia here again, because you already should have some test ETH.

For doing this task it is recommended to use [Remix](https://remix.ethereum.org/).

Once you created and deployed your contract you can check it with the validator smart contract.   
  
For making it you will need to follow the link to etherscan and pass your SС address to the function check(address \_yourContract) on the validator contract.

Getting the status **true** would mean that everything is done correctly.  
  
[Validator link](https://sepolia.etherscan.io/address/0x29da1560dd6abaaa522edb39e84f99191819f8c4#code).

What to hand in:  
  
You will have to provide the link to your verified contract.