2) Write a Smart Contract for understanding value types in Solidity

Ans.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract ValueTypeDemo {
  // State variables
  uint256 public uintVariable;
  int256 public intVariable;
  address public address Variable;
  bool public boolVariable;
  string public stringVariable;
  bytes32 public bytesVariable;
  // Constructor to initialize state variables
  constructor() {
     uintVariable = 10;
     intVariable = -5;
     addressVariable = msg.sender;
     boolVariable = true:
     stringVariable = "Hello, World!";
     bytesVariable = keccak256(abi.encodePacked("Solidity"));
  }
  // Function to update uintVariable
  function updateUint(uint256 newValue) public {
     uintVariable = newValue;
  }
  // Function to update intVariable
  function updateInt(int256 newValue) public {
     intVariable = newValue;
  }
  // Function to update addressVariable
```

```
function updateAddress(address newValue) public {
    addressVariable = newValue;
  }
  // Function to update boolVariable
  function updateBool(bool newValue) public {
    boolVariable = newValue:
  }
  // Function to update stringVariable
  function updateString(string memory newValue) public {
    stringVariable = newValue;
  }
  // Function to update bytesVariable
  function updateBytes(bytes32 newValue) public {
    bytesVariable = newValue;
  }
}
```

After compiling the Solidity smart contract, you'll get bytecode and ABI (Application Binary Interface) JSON.