ECC program

/\*\*

   \* @title ContractName

   \* @dev ContractDescription

   \* @custom:dev-run-script file\_path

   \*/

// SPDX-License-Identifier: MIT

pragma solidity 0.8.7;

contract SignatureUtility {

    //input a signature and determine the v, r and s values

    function splitSignature(bytes memory signature) public pure

        returns (uint8 v, bytes32 r, bytes32 s)

    {

        require(signature.length == 65);

        assembly {

            // first 32 bytes, after the length prefix.

            r := mload(add(signature, 32))

            // second 32 bytes.

            s := mload(add(signature, 64))

            // final byte (first byte of the next 32 bytes).

            v := byte(0, mload(add(signature, 96)))

        }

        return (v, r, s);

    }

    //input a message hash, v, r and s values and generate the public key

    function recoverPublicAddress (bytes32 messagehash, uint8 v, bytes32 r, bytes32 s) public pure

    returns (address sender) {

        return ecrecover(messagehash, v, r, s);

  }

}