Contract: SmartDoge

Compiler Version:	v0.6.12+commit.27d51765
Optimization Enabled:	No with 2000 runs
Other Settings:	default evmVersion

Creator:	0x175375105936D68Dfc50124D8fe0b6aED7982cab
Contract Address:	0x6e6D4ECE35EEd638A1153339F69E543B7ae5F776
Creation TX hash:	0x59b73a70f92974afe9dbd686aa7582257b17d804af0d2ba21507e2f7cf4a8a5b

Solidity Contract Code

```
pragma solidity ^0.6.12;

// SPDX-License-Identifier: Unlicensed
interface IERC20 {
    function totalSupply() external view returns (uint256);

    /**
     * @dev Returns the amount of tokens owned by `account`.
     */
    function balanceOf(address account) external view returns (uint256);

    /**
     * @dev Moves `amount` tokens from the caller's account to `recipient`.
     *
     * Returns a boolean value indicating whether the operation succeeded.
     *
     * Emits a {Transfer} event.
     */
    function transfer(address recipient, uint256 amount) external returns (bool);

    /**
     * @dev Returns the remaining number of tokens that `spender` will be
     * allowed to spend on behalf of `owner` through {transferFrom}. This is
     * zero by default.
     *
     * This value changes when {approve} or {transferFrom} are called.
     */
    function allowance(address owner, address spender) external view returns (uint256);
```

```
    /**
     * @dev Sets `amount` as the allowance of `spender` over the caller's tokens.
     *
     * Returns a boolean value indicating whether the operation succeeded.
     *
     * IMPORTANT: Beware that changing an allowance with this method brings the risk
     * that someone may use both the old and the new allowance by unfortunate
     * transaction ordering. One possible solution to mitigate this race
     * condition is to first reduce the spender's allowance to 0 and set the
     * desired value afterwards:
     * https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729
     *
     * Emits an {Approval} event.
     */
    function approve(address spender, uint256 amount) external returns (bool);

    /**
     * @dev Moves `amount` tokens from `sender` to `recipient` using the
     * allowance mechanism. `amount` is then deducted from the caller's
     * allowance.
     *
     * Returns a boolean value indicating whether the operation succeeded.
     *
     * Emits a {Transfer} event.
     */
    function transferFrom(
        address sender,
        address recipient,
        uint256 amount
    ) external returns (bool);

    /**
     * @dev Emitted when `value` tokens are moved from one account (`from`) to
     * another (`to`).
     *
     * Note that `value` may be zero.
     */
    event Transfer(address indexed from, address indexed to, uint256 value);

    /**
     * @dev Emitted when the allowance of a `spender` for an `owner` is set by
     * a call to {approve}. `value` is the new allowance.
     */
    event Approval(address indexed owner, address indexed spender, uint256 value);
}

/**
 * @dev Wrappers over Solidity's arithmetic operations with added overflow
 * checks.
```

```
 *
 * Arithmetic operations in Solidity wrap on overflow. This can easily result
 * in bugs, because programmers usually assume that an overflow raises an
 * error, which is the standard behavior in high level programming languages.
 * `SafeMath` restores this intuition by reverting the transaction when an
 * operation overflows.
 *
 * Using this library instead of the unchecked operations eliminates an entire
 * class of bugs, so it's recommended to use it always.
 */

library SafeMath {
    /**
     * @dev Returns the addition of two unsigned integers, reverting on
     * overflow.
     *
     * Counterpart to Solidity's `+` operator.
     *
     * Requirements:
     *
     * - Addition cannot overflow.
     */
    function add(uint256 a, uint256 b) internal pure returns (uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");

        return c;
    }

    /**
     * @dev Returns the subtraction of two unsigned integers, reverting on
     * overflow (when the result is negative).
     *
     * Counterpart to Solidity's `-` operator.
     *
     * Requirements:
     *
     * - Subtraction cannot overflow.
     */
    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        return sub(a, b, "SafeMath: subtraction overflow");
    }

    /**
     * @dev Returns the subtraction of two unsigned integers, reverting with custom message
on
     * overflow (when the result is negative).
     *
     * Counterpart to Solidity's `-` operator.
     *
```

```
     * Requirements:
     *
     * - Subtraction cannot overflow.
     */
    function sub(
        uint256 a,
        uint256 b,
        string memory errorMessage
    ) internal pure returns (uint256) {
        require(b <= a, errorMessage);
        uint256 c = a - b;

        return c;
    }

    /**
     * @dev Returns the multiplication of two unsigned integers, reverting on
     * overflow.
     *
     * Counterpart to Solidity's `*` operator.
     *
     * Requirements:
     *
     * - Multiplication cannot overflow.
     */
    function mul(uint256 a, uint256 b) internal pure returns (uint256) {
        // Gas optimization: this is cheaper than requiring 'a' not being zero, but the
        // benefit is lost if 'b' is also tested.
        // See: https://github.com/OpenZeppelin/openzeppelin-contracts/pull/522
        if (a == 0) {
            return 0;
        }

        uint256 c = a * b;
        require(c / a == b, "SafeMath: multiplication overflow");

        return c;
    }

    /**
     * @dev Returns the integer division of two unsigned integers. Reverts on
     * division by zero. The result is rounded towards zero.
     *
     * Counterpart to Solidity's `/` operator. Note: this function uses a
     * `revert` opcode (which leaves remaining gas untouched) while Solidity
     * uses an invalid opcode to revert (consuming all remaining gas).
     *
     * Requirements:
     *
     * - The divisor cannot be zero.
```

```
     */
    function div(uint256 a, uint256 b) internal pure returns (uint256) {
        return div(a, b, "SafeMath: division by zero");
    }

    /**
     * @dev Returns the integer division of two unsigned integers. Reverts with custom
message on
     * division by zero. The result is rounded towards zero.
     *
     * Counterpart to Solidity's `/` operator. Note: this function uses a
     * `revert` opcode (which leaves remaining gas untouched) while Solidity
     * uses an invalid opcode to revert (consuming all remaining gas).
     *
     * Requirements:
     *
     * - The divisor cannot be zero.
     */
    function div(
        uint256 a,
        uint256 b,
        string memory errorMessage
    ) internal pure returns (uint256) {
        require(b > 0, errorMessage);
        uint256 c = a / b;
        // assert(a == b * c + a % b); // There is no case in which this doesn't hold

        return c;
    }

    /**
     * @dev Returns the remainder of dividing two unsigned integers. (unsigned integer
modulo),
     * Reverts when dividing by zero.
     *
     * Counterpart to Solidity's `%` operator. This function uses a `revert`
     * opcode (which leaves remaining gas untouched) while Solidity uses an
     * invalid opcode to revert (consuming all remaining gas).
     *
     * Requirements:
     *
     * - The divisor cannot be zero.
     */
    function mod(uint256 a, uint256 b) internal pure returns (uint256) {
        return mod(a, b, "SafeMath: modulo by zero");
    }

    /**
     * @dev Returns the remainder of dividing two unsigned integers. (unsigned integer
modulo),
```

```
     * Reverts with custom message when dividing by zero.
     *
     * Counterpart to Solidity's `%` operator. This function uses a `revert`
     * opcode (which leaves remaining gas untouched) while Solidity uses an
     * invalid opcode to revert (consuming all remaining gas).
     *
     * Requirements:
     *
     * - The divisor cannot be zero.
     */
    function mod(
        uint256 a,
        uint256 b,
        string memory errorMessage
    ) internal pure returns (uint256) {
        require(b != 0, errorMessage);
        return a % b;
    }
}

abstract contract Context {
    function _msgSender() internal view virtual returns (address payable) {
        return msg.sender;
    }

    function _msgData() internal view virtual returns (bytes memory) {
        this; // silence state mutability warning without generating bytecode - see
https://github.com/ethereum/solidity/issues/2691
        return msg.data;
    }
}

/**
 * @dev Collection of functions related to the address type
 */
library Address {
    /**
     * @dev Returns true if `account` is a contract.
     *
     * [IMPORTANT]
     * ====
     * It is unsafe to assume that an address for which this function returns
     * false is an externally-owned account (EOA) and not a contract.
     *
     * Among others, `isContract` will return false for the following
     * types of addresses:
     *
     *  - an externally-owned account
     *  - a contract in construction
     *  - an address where a contract will be created
```

```
     *  - an address where a contract lived, but was destroyed
     * ====
     */
    function isContract(address account) internal view returns (bool) {
        // According to EIP-1052, 0x0 is the value returned for not-yet created accounts
        // and 0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470 is returned
        // for accounts without code, i.e. `keccak256('')`
        bytes32 codehash;
        bytes32 accountHash =
0xc5d2460186f7233c927e7db2dcc703c0e500b653ca82273b7bfad8045d85a470;
        // solhint-disable-next-line no-inline-assembly
        assembly {
            codehash := extcodehash(account)
        }
        return (codehash != accountHash && codehash != 0x0);
    }

    /**
     * @dev Replacement for Solidity's `transfer`: sends `amount` wei to
     * `recipient`, forwarding all available gas and reverting on errors.
     *
     * https://eips.ethereum.org/EIPS/eip-1884[EIP1884] increases the gas cost
     * of certain opcodes, possibly making contracts go over the 2300 gas limit
     * imposed by `transfer`, making them unable to receive funds via
     * `transfer`. {sendValue} removes this limitation.
     *
     * https://diligence.consensys.net/posts/2019/09/stop-using-soliditys-transfer-now/[Learn
more].
     *
     * IMPORTANT: because control is transferred to `recipient`, care must be
     * taken to not create reentrancy vulnerabilities. Consider using
     * {ReentrancyGuard} or the
     * https://solidity.readthedocs.io/en/v0.5.11/security-considerations.html#use-the-
checks-effects-interactions-pattern[checks-effects-interactions pattern].
     */
    function sendValue(address payable recipient, uint256 amount) internal {
        require(address(this).balance >= amount, "Address: insufficient balance");

        // solhint-disable-next-line avoid-low-level-calls, avoid-call-value
        (bool success, ) = recipient.call{value: amount}("");
        require(success, "Address: unable to send value, recipient may have reverted");
    }

    /**
     * @dev Performs a Solidity function call using a low level `call`. A
     * plain`call` is an unsafe replacement for a function call: use this
     * function instead.
     *
     * If `target` reverts with a revert reason, it is bubbled up by this
     * function (like regular Solidity function calls).
```

```
     *
     * Returns the raw returned data. To convert to the expected return value,
     * use https://solidity.readthedocs.io/en/latest/units-and-global-
variables.html?highlight=abi.decode#abi-encoding-and-decoding-functions[`abi.decode`].
     *
     * Requirements:
     *
     * - `target` must be a contract.
     * - calling `target` with `data` must not revert.
     *
     * _Available since v3.1._
     */
    function functionCall(address target, bytes memory data) internal returns (bytes memory)
{
        return functionCall(target, data, "Address: low-level call failed");
    }

    /**
     * @dev Same as {xref-Address-functionCall-address-bytes-}[`functionCall`], but with
     * `errorMessage` as a fallback revert reason when `target` reverts.
     *
     * _Available since v3.1._
     */
    function functionCall(
        address target,
        bytes memory data,
        string memory errorMessage
    ) internal returns (bytes memory) {
        return _functionCallWithValue(target, data, 0, errorMessage);
    }

    /**
     * @dev Same as {xref-Address-functionCall-address-bytes-}[`functionCall`],
     * but also transferring `value` wei to `target`.
     *
     * Requirements:
     *
     * - the calling contract must have an ETH balance of at least `value`.
     * - the called Solidity function must be `payable`.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, value, "Address: low-level call with value
failed");
    }
```

```
    /**
     * @dev Same as {xref-Address-functionCallWithValue-address-bytes-
uint256-}[`functionCallWithValue`], but
     * with `errorMessage` as a fallback revert reason when `target` reverts.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, "Address: insufficient balance for call");
        return _functionCallWithValue(target, data, value, errorMessage);
    }

    function _functionCallWithValue(
        address target,
        bytes memory data,
        uint256 weiValue,
        string memory errorMessage
    ) private returns (bytes memory) {
        require(isContract(target), "Address: call to non-contract");

        // solhint-disable-next-line avoid-low-level-calls
        (bool success, bytes memory returndata) = target.call{value: weiValue}(data);
        if (success) {
            return returndata;
        } else {
            // Look for revert reason and bubble it up if present
            if (returndata.length > 0) {
                // The easiest way to bubble the revert reason is using memory via assembly

                // solhint-disable-next-line no-inline-assembly
                assembly {
                    let returndata_size := mload(returndata)
                    revert(add(32, returndata), returndata_size)
                }
            } else {
                revert(errorMessage);
            }
        }
    }
}

/**
 * @dev Contract module which provides a basic access control mechanism, where
 * there is an account (an owner) that can be granted exclusive access to
```

```
 * specific functions.
 *
 * By default, the owner account will be the one that deploys the contract. This
 * can later be changed with {transferOwnership}.
 *
 * This module is used through inheritance. It will make available the modifier
 * `onlyOwner`, which can be applied to your functions to restrict their use to
 * the owner.
 */
contract Ownable is Context {
    address private _owner;
    address private _previousOwner;
    uint256 private _lockTime;

    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner);

    /**
     * @dev Initializes the contract setting the deployer as the initial owner.
     */
    constructor() internal {
        address msgSender = _msgSender();
        _owner = msgSender;
        emit OwnershipTransferred(address(0), msgSender);
    }

    /**
     * @dev Returns the address of the current owner.
     */
    function owner() public view returns (address) {
        return _owner;
    }

    /**
     * @dev Throws if called by any account other than the owner.
     */
    modifier onlyOwner() {
        require(_owner == _msgSender(), "Ownable: caller is not the owner");
        _;
    }

    /**
     * @dev Leaves the contract without owner. It will not be possible to call
     * `onlyOwner` functions anymore. Can only be called by the current owner.
     *
     * NOTE: Renouncing ownership will leave the contract without an owner,
     * thereby removing any functionality that is only available to the owner.
     */
    function renounceOwnership() public virtual onlyOwner {
        emit OwnershipTransferred(_owner, address(0));
        _owner = address(0);
```

```
    }

    /**
     * @dev Transfers ownership of the contract to a new account (`newOwner`).
     * Can only be called by the current owner.
     */
    function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        emit OwnershipTransferred(_owner, newOwner);
        _owner = newOwner;
    }

    function geUnlockTime() public view returns (uint256) {
        return _lockTime;
    }

    //Locks the contract for owner for the amount of time provided
    function lock(uint256 time) public virtual onlyOwner {
        _previousOwner = _owner;
        _owner = address(0);
        _lockTime = now + time;
        emit OwnershipTransferred(_owner, address(0));
    }

    //Unlocks the contract for owner when _lockTime is exceeds
    function unlock() public virtual {
        require(_previousOwner == msg.sender, "You don't have permission to unlock");
        require(now > _lockTime, "Contract is locked until 7 days");
        emit OwnershipTransferred(_owner, _previousOwner);
        _owner = _previousOwner;
    }
}

// pragma solidity >=0.5.0;

interface IUniswapV2Factory {
    event PairCreated(address indexed token0, address indexed token1, address pair, uint256);

    function feeTo() external view returns (address);

    function feeToSetter() external view returns (address);

    function getPair(address tokenA, address tokenB) external view returns (address pair);

    function allPairs(uint256) external view returns (address pair);

    function allPairsLength() external view returns (uint256);

    function createPair(address tokenA, address tokenB) external returns (address pair);
```

```
    function setFeeTo(address) external;

    function setFeeToSetter(address) external;
}

// pragma solidity >=0.5.0;

interface IUniswapV2Pair {
    event Approval(address indexed owner, address indexed spender, uint256 value);
    event Transfer(address indexed from, address indexed to, uint256 value);

    function name() external pure returns (string memory);

    function symbol() external pure returns (string memory);

    function decimals() external pure returns (uint8);

    function totalSupply() external view returns (uint256);

    function balanceOf(address owner) external view returns (uint256);

    function allowance(address owner, address spender) external view returns (uint256);

    function approve(address spender, uint256 value) external returns (bool);

    function transfer(address to, uint256 value) external returns (bool);

    function transferFrom(
        address from,
        address to,
        uint256 value
    ) external returns (bool);

    function DOMAIN_SEPARATOR() external view returns (bytes32);

    function PERMIT_TYPEHASH() external pure returns (bytes32);

    function nonces(address owner) external view returns (uint256);

    function permit(
        address owner,
        address spender,
        uint256 value,
        uint256 deadline,
        uint8 v,
        bytes32 r,
        bytes32 s
    ) external;

    event Mint(address indexed sender, uint256 amount0, uint256 amount1);
```

```
    event Burn(address indexed sender, uint256 amount0, uint256 amount1, address indexed to);
    event Swap(
        address indexed sender,
        uint256 amount0In,
        uint256 amount1In,
        uint256 amount0Out,
        uint256 amount1Out,
        address indexed to
    );
    event Sync(uint112 reserve0, uint112 reserve1);

    function MINIMUM_LIQUIDITY() external pure returns (uint256);

    function factory() external view returns (address);

    function token0() external view returns (address);

    function token1() external view returns (address);

    function getReserves()
        external
        view
        returns (
            uint112 reserve0,
            uint112 reserve1,
            uint32 blockTimestampLast
        );

    function price0CumulativeLast() external view returns (uint256);

    function price1CumulativeLast() external view returns (uint256);

    function kLast() external view returns (uint256);

    function mint(address to) external returns (uint256 liquidity);

    function burn(address to) external returns (uint256 amount0, uint256 amount1);

    function swap(
        uint256 amount0Out,
        uint256 amount1Out,
        address to,
        bytes calldata data
    ) external;

    function skim(address to) external;

    function sync() external;

    function initialize(address, address) external;
```

```
}

// pragma solidity >=0.6.2;

interface IUniswapV2Router01 {
    function factory() external pure returns (address);

    function WETH() external pure returns (address);

    function addLiquidity(
        address tokenA,
        address tokenB,
        uint256 amountADesired,
        uint256 amountBDesired,
        uint256 amountAMin,
        uint256 amountBMin,
        address to,
        uint256 deadline
    )
        external
        returns (
            uint256 amountA,
            uint256 amountB,
            uint256 liquidity
        );

    function addLiquidityETH(
        address token,
        uint256 amountTokenDesired,
        uint256 amountTokenMin,
        uint256 amountETHMin,
        address to,
        uint256 deadline
    )
        external
        payable
        returns (
            uint256 amountToken,
            uint256 amountETH,
            uint256 liquidity
        );

    function removeLiquidity(
        address tokenA,
        address tokenB,
        uint256 liquidity,
        uint256 amountAMin,
        uint256 amountBMin,
        address to,
        uint256 deadline
```

```
    ) external returns (uint256 amountA, uint256 amountB);

    function removeLiquidityETH(
        address token,
        uint256 liquidity,
        uint256 amountTokenMin,
        uint256 amountETHMin,
        address to,
        uint256 deadline
    ) external returns (uint256 amountToken, uint256 amountETH);

    function removeLiquidityWithPermit(
        address tokenA,
        address tokenB,
        uint256 liquidity,
        uint256 amountAMin,
        uint256 amountBMin,
        address to,
        uint256 deadline,
        bool approveMax,
        uint8 v,
        bytes32 r,
        bytes32 s
    ) external returns (uint256 amountA, uint256 amountB);

    function removeLiquidityETHWithPermit(
        address token,
        uint256 liquidity,
        uint256 amountTokenMin,
        uint256 amountETHMin,
        address to,
        uint256 deadline,
        bool approveMax,
        uint8 v,
        bytes32 r,
        bytes32 s
    ) external returns (uint256 amountToken, uint256 amountETH);

    function swapExactTokensForTokens(
        uint256 amountIn,
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external returns (uint256[] memory amounts);

    function swapTokensForExactTokens(
        uint256 amountOut,
        uint256 amountInMax,
        address[] calldata path,
```

```
        address to,
        uint256 deadline
    ) external returns (uint256[] memory amounts);

    function swapExactETHForTokens(
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external payable returns (uint256[] memory amounts);

    function swapTokensForExactETH(
        uint256 amountOut,
        uint256 amountInMax,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external returns (uint256[] memory amounts);

    function swapExactTokensForETH(
        uint256 amountIn,
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external returns (uint256[] memory amounts);

    function swapETHForExactTokens(
        uint256 amountOut,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external payable returns (uint256[] memory amounts);

    function quote(
        uint256 amountA,
        uint256 reserveA,
        uint256 reserveB
    ) external pure returns (uint256 amountB);

    function getAmountOut(
        uint256 amountIn,
        uint256 reserveIn,
        uint256 reserveOut
    ) external pure returns (uint256 amountOut);

    function getAmountIn(
        uint256 amountOut,
        uint256 reserveIn,
        uint256 reserveOut
```

```
    ) external pure returns (uint256 amountIn);

    function getAmountsOut(uint256 amountIn, address[] calldata path) external view returns
(uint256[] memory amounts);

    function getAmountsIn(uint256 amountOut, address[] calldata path) external view returns
(uint256[] memory amounts);
}

// pragma solidity >=0.6.2;

interface IUniswapV2Router02 is IUniswapV2Router01 {
    function removeLiquidityETHSupportingFeeOnTransferTokens(
        address token,
        uint256 liquidity,
        uint256 amountTokenMin,
        uint256 amountETHMin,
        address to,
        uint256 deadline
    ) external returns (uint256 amountETH);

    function removeLiquidityETHWithPermitSupportingFeeOnTransferTokens(
        address token,
        uint256 liquidity,
        uint256 amountTokenMin,
        uint256 amountETHMin,
        address to,
        uint256 deadline,
        bool approveMax,
        uint8 v,
        bytes32 r,
        bytes32 s
    ) external returns (uint256 amountETH);

    function swapExactTokensForTokensSupportingFeeOnTransferTokens(
        uint256 amountIn,
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external;

    function swapExactETHForTokensSupportingFeeOnTransferTokens(
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external payable;

    function swapExactTokensForETHSupportingFeeOnTransferTokens(
```

```
        uint256 amountIn,
        uint256 amountOutMin,
        address[] calldata path,
        address to,
        uint256 deadline
    ) external;
}

contract SmartDoge is Context, IERC20, Ownable {
    using SafeMath for uint256;
    using Address for address;

    mapping(address => uint256) private _rOwned;
    mapping(address => uint256) private _tOwned;
    mapping(address => mapping(address => uint256)) private _allowances;

    mapping(address => bool) private _isExcludedFromFee;

    mapping(address => bool) private _isExcluded;
    address[] private _excluded;

    uint256 private constant MAX = ~uint256(0);
    uint256 private _tTotal = 1000000000 * 10**6 * 10**9;
    uint256 private _rTotal = (MAX - (MAX % _tTotal));
    uint256 private _tFeeTotal;

    string private _name = "SmartDoge";
    string private _symbol = "SMART";
    uint8 private _decimals = 9;

    uint256 public _taxFee = 5;
    uint256 private _previousTaxFee = _taxFee;

    uint256 public _liquidityFee = 5;
    uint256 private _previousLiquidityFee = _liquidityFee;

    IUniswapV2Router02 public immutable uniswapV2Router;
    address public immutable uniswapV2Pair;

    bool inSwapAndLiquify;
    bool public swapAndLiquifyEnabled = true;

    uint256 public _maxTxAmount = 5000000 * 10**6 * 10**9;
    uint256 private numTokensSellToAddToLiquidity = 500000 * 10**6 * 10**9;

    event MinTokensBeforeSwapUpdated(uint256 minTokensBeforeSwap);
    event SwapAndLiquifyEnabledUpdated(bool enabled);
    event SwapAndLiquify(uint256 tokensSwapped, uint256 ethReceived, uint256
tokensIntoLiqudity);
```

```
    modifier lockTheSwap() {
        inSwapAndLiquify = true;
        _;
        inSwapAndLiquify = false;
    }

    constructor() public {
        _rOwned[_msgSender()] = _rTotal;

        IUniswapV2Router02 _uniswapV2Router =
IUniswapV2Router02(0x5d0bF8d8c8b054080E2131D8b260a5c6959411B8);
        // Create a uniswap pair for this new token
        uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(
            address(this),
            _uniswapV2Router.WETH()
        );

        // set the rest of the contract variables
        uniswapV2Router = _uniswapV2Router;

        //exclude owner and this contract from fee
        _isExcludedFromFee[owner()] = true;
        _isExcludedFromFee[address(this)] = true;

        emit Transfer(address(0), _msgSender(), _tTotal);
    }

    function name() public view returns (string memory) {
        return _name;
    }

    function symbol() public view returns (string memory) {
        return _symbol;
    }

    function decimals() public view returns (uint8) {
        return _decimals;
    }

    function totalSupply() public view override returns (uint256) {
        return _tTotal;
    }

    function balanceOf(address account) public view override returns (uint256) {
        if (_isExcluded[account]) return _tOwned[account];
        return tokenFromReflection(_rOwned[account]);
    }

    function transfer(address recipient, uint256 amount) public override returns (bool) {
        _transfer(_msgSender(), recipient, amount);
```

```
        return true;
    }

    function allowance(address owner, address spender) public view override returns (uint256)
{
        return _allowances[owner][spender];
    }

    function approve(address spender, uint256 amount) public override returns (bool) {
        _approve(_msgSender(), spender, amount);
        return true;
    }

    function transferFrom(
        address sender,
        address recipient,
        uint256 amount
    ) public override returns (bool) {
        _transfer(sender, recipient, amount);
        _approve(
            sender,
            _msgSender(),
            _allowances[sender][_msgSender()].sub(amount, "ERC20: transfer amount exceeds
allowance")
        );
        return true;
    }

    function increaseAllowance(address spender, uint256 addedValue) public virtual returns
(bool) {
        _approve(_msgSender(), spender, _allowances[_msgSender()][spender].add(addedValue));
        return true;
    }

    function decreaseAllowance(address spender, uint256 subtractedValue) public virtual
returns (bool) {
        _approve(
            _msgSender(),
            spender,
            _allowances[_msgSender()][spender].sub(subtractedValue, "ERC20: decreased
allowance below zero")
        );
        return true;
    }

    function isExcludedFromReward(address account) public view returns (bool) {
        return _isExcluded[account];
    }

    function totalFees() public view returns (uint256) {
```

```
        return _tFeeTotal;
    }

    function deliver(uint256 tAmount) public {
        address sender = _msgSender();
        require(!_isExcluded[sender], "Excluded addresses cannot call this function");
        (uint256 rAmount, , , , , ) = _getValues(tAmount);
        _rOwned[sender] = _rOwned[sender].sub(rAmount);
        _rTotal = _rTotal.sub(rAmount);
        _tFeeTotal = _tFeeTotal.add(tAmount);
    }

    function reflectionFromToken(uint256 tAmount, bool deductTransferFee) public view returns
(uint256) {
        require(tAmount <= _tTotal, "Amount must be less than supply");
        if (!deductTransferFee) {
            (uint256 rAmount, , , , , ) = _getValues(tAmount);
            return rAmount;
        } else {
            (, uint256 rTransferAmount, , , , ) = _getValues(tAmount);
            return rTransferAmount;
        }
    }

    function tokenFromReflection(uint256 rAmount) public view returns (uint256) {
        require(rAmount <= _rTotal, "Amount must be less than total reflections");
        uint256 currentRate = _getRate();
        return rAmount.div(currentRate);
    }

    function excludeFromReward(address account) public onlyOwner {
        // require(account != 0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D, 'We can not exclude
Uniswap router.');
        require(!_isExcluded[account], "Account is already excluded");
        if (_rOwned[account] > 0) {
            _tOwned[account] = tokenFromReflection(_rOwned[account]);
        }
        _isExcluded[account] = true;
        _excluded.push(account);
    }

    function includeInReward(address account) external onlyOwner {
        require(_isExcluded[account], "Account is already excluded");
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_excluded[i] == account) {
                _excluded[i] = _excluded[_excluded.length - 1];
                _tOwned[account] = 0;
                _isExcluded[account] = false;
                _excluded.pop();
                break;
```

```
        }
    }
}

function _transferBothExcluded(
    address sender,
    address recipient,
    uint256 tAmount
) private {
    (
        uint256 rAmount,
        uint256 rTransferAmount,
        uint256 rFee,
        uint256 tTransferAmount,
        uint256 tFee,
        uint256 tLiquidity
    ) = _getValues(tAmount);
    _tOwned[sender] = _tOwned[sender].sub(tAmount);
    _rOwned[sender] = _rOwned[sender].sub(rAmount);
    _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount);
    _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount);
    _takeLiquidity(tLiquidity);
    _reflectFee(rFee, tFee);
    emit Transfer(sender, recipient, tTransferAmount);
}

function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}

function setTaxFeePercent(uint256 taxFee) external onlyOwner {
    _taxFee = taxFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner {
    _liquidityFee = liquidityFee;
}

function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(10**2);
}

function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

```
    //to recieve ETH from uniswapV2Router when swaping
    receive() external payable {}

    function _reflectFee(uint256 rFee, uint256 tFee) private {
        _rTotal = _rTotal.sub(rFee);
        _tFeeTotal = _tFeeTotal.add(tFee);
    }

    function _getValues(uint256 tAmount)
        private
        view
        returns (
            uint256,
            uint256,
            uint256,
            uint256,
            uint256,
            uint256
        )
    {
        (uint256 tTransferAmount, uint256 tFee, uint256 tLiquidity) = _getTValues(tAmount);
        (uint256 rAmount, uint256 rTransferAmount, uint256 rFee) = _getRValues(tAmount, tFee,
tLiquidity, _getRate());
        return (rAmount, rTransferAmount, rFee, tTransferAmount, tFee, tLiquidity);
    }

    function _getTValues(uint256 tAmount)
        private
        view
        returns (
            uint256,
            uint256,
            uint256
        )
    {
        uint256 tFee = calculateTaxFee(tAmount);
        uint256 tLiquidity = calculateLiquidityFee(tAmount);
        uint256 tTransferAmount = tAmount.sub(tFee).sub(tLiquidity);
        return (tTransferAmount, tFee, tLiquidity);
    }

    function _getRValues(
        uint256 tAmount,
        uint256 tFee,
        uint256 tLiquidity,
        uint256 currentRate
    )
        private
        pure
```

```
        returns (
            uint256,
            uint256,
            uint256
        )
    {
        uint256 rAmount = tAmount.mul(currentRate);
        uint256 rFee = tFee.mul(currentRate);
        uint256 rLiquidity = tLiquidity.mul(currentRate);
        uint256 rTransferAmount = rAmount.sub(rFee).sub(rLiquidity);
        return (rAmount, rTransferAmount, rFee);
    }

    function _getRate() private view returns (uint256) {
        (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
        return rSupply.div(tSupply);
    }

    function _getCurrentSupply() private view returns (uint256, uint256) {
        uint256 rSupply = _rTotal;
        uint256 tSupply = _tTotal;
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return
(_rTotal, _tTotal);
            rSupply = rSupply.sub(_rOwned[_excluded[i]]);
            tSupply = tSupply.sub(_tOwned[_excluded[i]]);
        }
        if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
        return (rSupply, tSupply);
    }

    function _takeLiquidity(uint256 tLiquidity) private {
        uint256 currentRate = _getRate();
        uint256 rLiquidity = tLiquidity.mul(currentRate);
        _rOwned[address(this)] = _rOwned[address(this)].add(rLiquidity);
        if (_isExcluded[address(this)]) _tOwned[address(this)] =
_tOwned[address(this)].add(tLiquidity);
    }

    function calculateTaxFee(uint256 _amount) private view returns (uint256) {
        return _amount.mul(_taxFee).div(10**2);
    }

    function calculateLiquidityFee(uint256 _amount) private view returns (uint256) {
        return _amount.mul(_liquidityFee).div(10**2);
    }

    function removeAllFee() private {
        if (_taxFee == 0 && _liquidityFee == 0) return;
```

```
        _previousTaxFee = _taxFee;
        _previousLiquidityFee = _liquidityFee;

        _taxFee = 0;
        _liquidityFee = 0;
    }

    function restoreAllFee() private {
        _taxFee = _previousTaxFee;
        _liquidityFee = _previousLiquidityFee;
    }

    function isExcludedFromFee(address account) public view returns (bool) {
        return _isExcludedFromFee[account];
    }

    function _approve(
        address owner,
        address spender,
        uint256 amount
    ) private {
        require(owner != address(0), "ERC20: approve from the zero address");
        require(spender != address(0), "ERC20: approve to the zero address");

        _allowances[owner][spender] = amount;
        emit Approval(owner, spender, amount);
    }

    function _transfer(
        address from,
        address to,
        uint256 amount
    ) private {
        require(from != address(0), "ERC20: transfer from the zero address");
        require(to != address(0), "ERC20: transfer to the zero address");
        require(amount > 0, "Transfer amount must be greater than zero");
        if (from != owner() && to != owner())
            require(amount <= _maxTxAmount, "Transfer amount exceeds the maxTxAmount.");

        // is the token balance of this contract address over the min number of
        // tokens that we need to initiate a swap + liquidity lock?
        // also, don't get caught in a circular liquidity event.
        // also, don't swap & liquify if sender is uniswap pair.
        uint256 contractTokenBalance = balanceOf(address(this));

        if (contractTokenBalance >= _maxTxAmount) {
            contractTokenBalance = _maxTxAmount;
        }

        bool overMinTokenBalance = contractTokenBalance >= numTokensSellToAddToLiquidity;
```

```
        if (overMinTokenBalance && !inSwapAndLiquify && from != uniswapV2Pair &&
swapAndLiquifyEnabled) {
            contractTokenBalance = numTokensSellToAddToLiquidity;
            //add liquidity
            swapAndLiquify(contractTokenBalance);
        }

        //indicates if fee should be deducted from transfer
        bool takeFee = true;

        //if any account belongs to _isExcludedFromFee account then remove the fee
        if (_isExcludedFromFee[from] || _isExcludedFromFee[to]) {
            takeFee = false;
        }

        //transfer amount, it will take tax, burn, liquidity fee
        _tokenTransfer(from, to, amount, takeFee);
    }

    function swapAndLiquify(uint256 contractTokenBalance) private lockTheSwap {
        // split the contract balance into halves
        uint256 half = contractTokenBalance.div(2);
        uint256 otherHalf = contractTokenBalance.sub(half);

        // capture the contract's current ETH balance.
        // this is so that we can capture exactly the amount of ETH that the
        // swap creates, and not make the liquidity event include any ETH that
        // has been manually sent to the contract
        uint256 initialBalance = address(this).balance;

        // swap tokens for ETH
        swapTokensForEth(half); // <- this breaks the ETH -> HATE swap when swap+liquify is
triggered

        // how much ETH did we just swap into?
        uint256 newBalance = address(this).balance.sub(initialBalance);

        // add liquidity to uniswap
        addLiquidity(otherHalf, newBalance);

        emit SwapAndLiquify(half, newBalance, otherHalf);
    }

    function swapTokensForEth(uint256 tokenAmount) private {
        // generate the uniswap pair path of token -> weth
        address[] memory path = new address[](2);
        path[0] = address(this);
        path[1] = uniswapV2Router.WETH();

        _approve(address(this), address(uniswapV2Router), tokenAmount);
```

```
        // make the swap
        uniswapV2Router.swapExactTokensForETHSupportingFeeOnTransferTokens(
            tokenAmount,
            0, // accept any amount of ETH
            path,
            address(this),
            block.timestamp
        );
    }

    function addLiquidity(uint256 tokenAmount, uint256 ethAmount) private {
        // approve token transfer to cover all possible scenarios
        _approve(address(this), address(uniswapV2Router), tokenAmount);

        // add the liquidity
        uniswapV2Router.addLiquidityETH{value: ethAmount}(
            address(this),
            tokenAmount,
            0, // slippage is unavoidable
            0, // slippage is unavoidable
            owner(),
            block.timestamp
        );
    }

    //this method is responsible for taking all fee, if takeFee is true
    function _tokenTransfer(
        address sender,
        address recipient,
        uint256 amount,
        bool takeFee
    ) private {
        if (!takeFee) removeAllFee();

        if (_isExcluded[sender] && !_isExcluded[recipient]) {
            _transferFromExcluded(sender, recipient, amount);
        } else if (!_isExcluded[sender] && _isExcluded[recipient]) {
            _transferToExcluded(sender, recipient, amount);
        } else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
            _transferStandard(sender, recipient, amount);
        } else if (_isExcluded[sender] && _isExcluded[recipient]) {
            _transferBothExcluded(sender, recipient, amount);
        } else {
            _transferStandard(sender, recipient, amount);
        }

        if (!takeFee) restoreAllFee();
    }
```

```
    function _transferStandard(
        address sender,
        address recipient,
        uint256 tAmount
    ) private {
        (
            uint256 rAmount,
            uint256 rTransferAmount,
            uint256 rFee,
            uint256 tTransferAmount,
            uint256 tFee,
            uint256 tLiquidity
        ) = _getValues(tAmount);
        _rOwned[sender] = _rOwned[sender].sub(rAmount);
        _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount);
        _takeLiquidity(tLiquidity);
        _reflectFee(rFee, tFee);
        emit Transfer(sender, recipient, tTransferAmount);
    }

    function _transferToExcluded(
        address sender,
        address recipient,
        uint256 tAmount
    ) private {
        (
            uint256 rAmount,
            uint256 rTransferAmount,
            uint256 rFee,
            uint256 tTransferAmount,
            uint256 tFee,
            uint256 tLiquidity
        ) = _getValues(tAmount);
        _rOwned[sender] = _rOwned[sender].sub(rAmount);
        _tOwned[recipient] = _tOwned[recipient].add(tTransferAmount);
        _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount);
        _takeLiquidity(tLiquidity);
        _reflectFee(rFee, tFee);
        emit Transfer(sender, recipient, tTransferAmount);
    }

    function _transferFromExcluded(
        address sender,
        address recipient,
        uint256 tAmount
    ) private {
        (
            uint256 rAmount,
            uint256 rTransferAmount,
            uint256 rFee,
```

```
            uint256 tTransferAmount,
            uint256 tFee,
            uint256 tLiquidity
        ) = _getValues(tAmount);
        _tOwned[sender] = _tOwned[sender].sub(tAmount);
        _rOwned[sender] = _rOwned[sender].sub(rAmount);
        _rOwned[recipient] = _rOwned[recipient].add(rTransferAmount);
        _takeLiquidity(tLiquidity);
        _reflectFee(rFee, tFee);
        emit Transfer(sender, recipient, tTransferAmount);
    }
}
```

Contract ABI

```
[
    {
      "inputs": [],
      "stateMutability": "nonpayable",
      "type": "constructor"
    },
    {
      "anonymous": false,
      "inputs": [
        {
          "indexed": true,
          "internalType": "address",
          "name": "owner",
          "type": "address"
        },
        {
          "indexed": true,
          "internalType": "address",
          "name": "spender",
          "type": "address"
        },
        {
          "indexed": false,
          "internalType": "uint256",
          "name": "value",
          "type": "uint256"
        }
      ],
      "name": "Approval",
      "type": "event"
    },
    {
      "anonymous": false,
```

```
    "inputs": [
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "minTokensBeforeSwap",
        "type": "uint256"
      }
    ],
    "name": "MinTokensBeforeSwapUpdated",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "previousOwner",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "newOwner",
        "type": "address"
      }
    ],
    "name": "OwnershipTransferred",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "tokensSwapped",
        "type": "uint256"
      },
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "ethReceived",
        "type": "uint256"
      },
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "tokensIntoLiqudity",
        "type": "uint256"
```

```
      }
    ],
    "name": "SwapAndLiquify",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": false,
        "internalType": "bool",
        "name": "enabled",
        "type": "bool"
      }
    ],
    "name": "SwapAndLiquifyEnabledUpdated",
    "type": "event"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "address",
        "name": "from",
        "type": "address"
      },
      {
        "indexed": true,
        "internalType": "address",
        "name": "to",
        "type": "address"
      },
      {
        "indexed": false,
        "internalType": "uint256",
        "name": "value",
        "type": "uint256"
      }
    ],
    "name": "Transfer",
    "type": "event"
  },
  {
    "inputs": [],
    "name": "_liquidityFee",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
```

```
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "_maxTxAmount",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "_taxFee",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "owner",
        "type": "address"
      },
      {
        "internalType": "address",
        "name": "spender",
        "type": "address"
      }
    ],
    "name": "allowance",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
```

```
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "spender",
        "type": "address"
      },
      {
        "internalType": "uint256",
        "name": "amount",
        "type": "uint256"
      }
    ],
    "name": "approve",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "balanceOf",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
```

```
    "inputs": [],
    "name": "decimals",
    "outputs": [
      {
        "internalType": "uint8",
        "name": "",
        "type": "uint8"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "spender",
        "type": "address"
      },
      {
        "internalType": "uint256",
        "name": "subtractedValue",
        "type": "uint256"
      }
    ],
    "name": "decreaseAllowance",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "tAmount",
        "type": "uint256"
      }
    ],
    "name": "deliver",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
```

```
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "excludeFromFee",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "excludeFromReward",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "geUnlockTime",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "includeInFee",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
```

```
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "includeInReward",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "spender",
        "type": "address"
      },
      {
        "internalType": "uint256",
        "name": "addedValue",
        "type": "uint256"
      }
    ],
    "name": "increaseAllowance",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "isExcludedFromFee",
    "outputs": [
      {
        "internalType": "bool",
```

```
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "account",
        "type": "address"
      }
    ],
    "name": "isExcludedFromReward",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "time",
        "type": "uint256"
      }
    ],
    "name": "lock",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "name",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "view",
```

```
    "type": "function"
  },
  {
    "inputs": [],
    "name": "owner",
    "outputs": [
      {
        "internalType": "address",
        "name": "",
        "type": "address"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "tAmount",
        "type": "uint256"
      },
      {
        "internalType": "bool",
        "name": "deductTransferFee",
        "type": "bool"
      }
    ],
    "name": "reflectionFromToken",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "renounceOwnership",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
```

```
        "name": "liquidityFee",
        "type": "uint256"
      }
    ],
    "name": "setLiquidityFeePercent",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "maxTxPercent",
        "type": "uint256"
      }
    ],
    "name": "setMaxTxPercent",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "bool",
        "name": "_enabled",
        "type": "bool"
      }
    ],
    "name": "setSwapAndLiquifyEnabled",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "taxFee",
        "type": "uint256"
      }
    ],
    "name": "setTaxFeePercent",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
```

```
    "name": "swapAndLiquifyEnabled",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "symbol",
    "outputs": [
      {
        "internalType": "string",
        "name": "",
        "type": "string"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "rAmount",
        "type": "uint256"
      }
    ],
    "name": "tokenFromReflection",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "totalFees",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
```

```
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "totalSupply",
    "outputs": [
      {
        "internalType": "uint256",
        "name": "",
        "type": "uint256"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "recipient",
        "type": "address"
      },
      {
        "internalType": "uint256",
        "name": "amount",
        "type": "uint256"
      }
    ],
    "name": "transfer",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "sender",
        "type": "address"
      },
```

```
      {
        "internalType": "address",
        "name": "recipient",
        "type": "address"
      },
      {
        "internalType": "uint256",
        "name": "amount",
        "type": "uint256"
      }
    ],
    "name": "transferFrom",
    "outputs": [
      {
        "internalType": "bool",
        "name": "",
        "type": "bool"
      }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [
      {
        "internalType": "address",
        "name": "newOwner",
        "type": "address"
      }
    ],
    "name": "transferOwnership",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "uniswapV2Pair",
    "outputs": [
      {
        "internalType": "address",
        "name": "",
        "type": "address"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [],
```

```
      "name": "uniswapV2Router",
      "outputs": [
        {
          "internalType": "contract IUniswapV2Router02",
          "name": "",
          "type": "address"
        }
      ],
      "stateMutability": "view",
      "type": "function"
    },
    {
      "inputs": [],
      "name": "unlock",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    },
    {
      "stateMutability": "payable",
      "type": "receive"
    }
  ]
```


Deployed Bytecode


```
"0x608060405260043610610026e5760003560e01c80635342acb411610153578063a457c2d7116100cb578063d543
dbeb1161007f578063dd62ed3e11610064578063dd62ed3e14610837578063ea2f0b3714610872578063f2fde38b1
46108a557610275565b8063d543dbeb146107e3578063dd4670641461080d57610275565b8063a9059cbb116100b0
578063a9059cbb14610769578063b6c52324146107a2578063c49b9a80146107b757610275565b8063a457c2d7146
1071b578063a69df4b5146107545761027565b80637d1db4a5116101225780638da5cb5b116101075780638da5cb
5b146106c75780638ee88c53146106dc57806395d89b4114610706576102757d1db4a51461067f5780638
8f8202014610694576102757d10275565b80635342acb4146105ef5780636bc87c3a146106225780637a082311461063757
8063715018a61461066a576102757d10275565b80633685d419116101e6578063437823ec116101b557806349bd5a5e11610
19a57806349bd5a5e146105925780634a74bb02146105a757806352390c02146105bc576102757d10275565b8063437823ec
1461052d5780634549b039146105605761027565b80633685d4191461048257806339509351146104b557806633b1
24fe7146104ee5780633bd5d173146105035761027565b80631694505e1161023d57806323b872dd116102257806
6323b872dd146103ea5780632d8381191461042d578063313ce567146104575761027565b80631694505e146103a
457806318160ddd146103d557610275565b8063061c82d01461027a57806306fdde03146102a6578063095ea7b314
6103305780631314a9d1461037d576102755b3661027557005b600080fd5b34801561028657600080fd5b50610
2a460048036036020811015610290d576100080fd5b50356108d8565b005b3480156102b257600080fd5b506102bb61
0947565b60408051602080825283518183015283519192839290830191850190808383600005b838110156102f5578
1810151838201526020016102dd565b50505050905090810190601f16801561032257808203805160018360200361
01000a031916815260200191505b509250505060405180910390f35b34801561033c57600080fd5b5061036960048
0360360408110156103535760080fd5b506001600160a01b0381351690602001356109dd565b6040805191151582
5251908190036020019f35b34801561038957600080fd5b506103926109fb565b60408051918252519081900360200
190f35b3480156103b057600080fd5b506103b9610a01565b60408051600160a01b0390921682525190819003602
0360200190f35b3480156103e157600080fd5b50610392610a25565b3480156103f657600080fd5b5061036960048
036036081101561040d57600080fd5b506001600160a01b0381358116916020810135909116906040013561a2b
```

565b34801561043957600080fd5b506103926004803603602081101561045057600080fd5b5035610ab2565b34801
561046357600080fd5b5061046c610b14565b6040805160ff909216825251908190036020019 0f35b34801561048e
57600080fd5b506102a46004803603602081101561 04a557600080fd5b5035600160016 0a01b0316610b1d565b348
0156104c157600080fd5b506103696004803603604081101561 04d857600080fd5b506001600160a01b0381351690
60200135610d0a565b3480156104fa57600080fd5b50610392610d58565b34801561050f57600080fd5b506102a46
00480360360208110156105265760 0080fd5b5035610d5e565b34801561053957600080fd5b506102a46004803603
602081101561055057600080fd5b5035600160016 0a01b0316610e38565b34801561056c57600080fd5b506103926
00480360360408110156105835760 0080fd5b508035906020013515156 10ec6565b34801561059e57600080fd5b50
6103b9610f58565b3480156105b357600080fd5b50610369610f7c565b3480156105c857600080fd5b506102a4600
480360360208110156105df57600080fd5b5035600160016 0a01b0316610f8a565b3480156105fb57600080fd5b50
610369600480360360208110156106125760 0080fd5b5035600160016 0a01b031661112f565b34801561062e57600
080fd5b5061039261114d565b34801561064357600080fd5b506103926004803603602081101561065a57600080fd
5b5035600160016 0a01b0316611153565b34801561067657600080fd5b506102a46111b5565b34801561068b57600
080fd5b506103926112765 65b3480156106a057600080fd5b506103696004803603602081101561 06b757600080fd
5b5035600160016 0a01b031661127c565b3480156106d357600080fd5b506103b961129a565b3480156106e857600
080fd5b506102a46004803603602081101561 06ff57600080fd5b50356112a9565b34801561071257 5600080fd5b50
6102bb611318565b34801561072757600080fd5b506103696004803603604081101561073e57600080fd5b5060016
00160a01b038135169060200135611379565b34801561076057600080fd5b506102a46113e1565b34801561077557
600080fd5b506103696004803603604081101561078c57600080fd5b506001600160a01b038135169060200135611
4ee565b3480156107ae57600080fd5b506103926115025 65b3480156107c357600080fd5b506102a4600480360360
208110156107da57600080fd5b5035151561150856 5b3480156107ef57600080fd5b506102a460048036036020811
0156108065760 0080fd5b50356115de565b34801561081957600080fd5b506102a46004803603602081101561083
57600080fd5b503561166e565b34801561084357600080fd5b506103926004803603604081101561085a57600080f
d5b506001600160a01b0381358116916020013516611 73d565b34801561087e57600080fd5b506102a46004803603
602081101561089557600080fd5b5035600160016 0a01b0316611768565b3480156108b157600080fd5b506102a46
00480360360208110156108c857600080fd5b5035600160016 0a01b03166117f3565b6108e061190a565b60005460
01600160a01b039081169116146109425760 40805162461bcd60e51b815260206004820181905260248201527f4f7
76e61626c653a2063616c6c6572206973206e6f742074686 5206f776e657260448201529051908190036064019 0fd
5b600f55565b600c805460408051602060 1f6002600019610100600188161502019095169490940493840181900 48
1028201810190925282815260609390929091830182828015610 9d35780601f106109a857610100808354040283 52
91602001916109d3565b8201919060005260206000209 05b8154815290600101906020018083116109b6578290036
01f168201915b505050505090509056 5b60006109f16109ea61190a565b848461190e565b5060015b92915050565b
600b5490565b7f0000000000000000000000000000000000000000000000000000000000000000000000000081565b600954905
65b6000610a388484846119fa565b610aa884610a4461190a565b610aa38560405180606001604052806028815260
2001612a876028913960016001600a01b038a1660009081526005602052604081209 0610a8261190a565b600160016
0a01b031681526020810191909152604001600020549190611c40565b61190e565b5060019392505050565b600060
0a54821115610af55760405162461bcd60e51b815260040180806020018281038252602a8152602001806129cc602
a9139604001915050604051809 10390fd5b6000610aff611cd7565b9050610b0b8382611cfa565b9150505b919050
565b600e5460ff1690565b610b2561190a565b60005460016001600a01b039081169116146 10b87576040805162461
bcd60e51b815260206004820181905260248201527f4f776e61626c653a2063616c6c6572206973206e6f74207468
65206f776e657260448201529051908190036064019 0fd5b6001600160a01b0381166000908152600760205260409
0205460ff16610bf4576040805162461bcd60e51b815260206004820152601b60248201527f4163636f756e742069
7320616c72656164792065636c7564656400000000006044820152905190819003606401 90fd5b60005b6008548
11015610d065781600160016 0a01b0316600882815481 10610c1857fe5b6000918252602090912001546001600160
a01b03161415610cfe576008805460001981019081 10610c4557fe5b6000918252602090912001546008805460016
00160a01b039092169183908110610c6b57fe5b6000918252602080832091909101805473ffffffffffffffffffff
ffffffffffffffffff19166001600160a01b0394851617905591841681526004825260408082208290556007909
25220805460ff191690556008805480610cca57fe5b600082815260209020810160001990810180547 3ffffffffff
ffffffffffffffffffffffff191690550190556 10d06565b600101610bf7565b5050565b60006109f1610d1
761190a565b84610aa38560056000610d2861190a565b6001600160a01b0390811682526020808301939093526040

9182016000908120918c168152925290205490611d43565b600f5481565b6000610d6861190a565b6001600160a01
b038116600090815260076020526040902054909150 60ff1615610dc35760405162461bcd60e51b81526004018080
6020018281038252602c815260200180612b21602c91396040019150506040518091039 0fd5b6000610dce83611d9
d565b505050506001600160a01b038416600090815260036020526040902054919250610dfa91905082611dec565b
6001600160a01b038316600090815260036020526040902055600a54610e209082611dec565b600a55600b54610e3
09084611d43565b600b55505050565b610e4061190a565b60005460016001600160a01b0390811691161461 0ea2576040
805162461bcd60e51b815260206004820181905260248201527f4f776e61626c653a2063616c6c65722069732061 6e6
f7420746865206f776e657260448201529051908190036064019 0fd5b6001600160a01b0316600090815260066020
5260409020805460ff1916600117905556 5b60006009548311156 10f1f576040805162461bcd60e51b81526020600
4820152601f60248201527f416d6f756e74206d757374206265206c657373207468616e20737570706c7900604482
01529051908190036064019 0fd5b81610f3e576000610f2f84611d9d565b509395506109f594505050505050565b600
0610f4984611d9d565b509295506109f594505050505050565b7f0000000000000000000000000000000000000000000000000
0000000000000000000081565b60135461010090046 0ff1681565b610f9261190a565b60005460016001600160a01b0
3908116911614610ff4576040805162461bcd60e51b815260206004820181905260248201527f4f776e61626c653a
2063616c6c657220697320616c726561647920657 8636c756465640000000060448201529
05190819003606401 90fd5b6001600160a01b038116600090815260036020526040902054156110bc576001600160
a01b038116600090815260036020526040902054611 0a290610ab2565b6001600160a01b0382166600090815260046
0205260409020555b6001600160a01b0316600081815260076020526040812080 5460ff1916600190811790915560
0880549182018155909 1527ff3f7a9fe364faab93b216da50a3214154f22a0a2b415b23a84c8169e8b636ee301805
473fffffffffffffffffffffffffffffffffffffff1916909117905556 5b6001600160a01b031660009081526006
60205260409020546 0ff1690565b60115481565b6001600160a01b038116600090815260076020526040812054 60f
f161561119357506001600160a01b038116600090815260046020526040902054610b0f565b6001600160a01b0382
166000908152600360205260409020546109f590610ab2565b6111bd61190a565b60005460016001600160a01b0390811
691161461121f576040805162461bcd60e51b815260206004820181905260248201527f4f776e61626c653a206361
6c6c6572206973206e6f7420746865206f776e657260448201529051908190036064019 0fd5b60008054604051600
1600160a01b03909116907f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0908390
a36000805473ffffffffffffffffffffffffffffffffffffffff19169055565b60145481565b6001600160a01b031
660009081526007602052604090205460ff1690565b60005460016001600160a01b031690565b6112b161190a565b6000
546001600160a01b0390811691161461131357604080516 2461bcd60e51b815260206004820181905260248201527
f4f776e61626c653a2063616c6c6572206973206e6f7420746865206f776e657260448201529051908190036064019
90fd5b601155565b600d805460408051602060 1f6002600019610100600188161502019095169490940493840181 9
00481028201810190925282815260609390929091830182820156109d35780601f106109a8576101008083540402
835291602001916109d3565b60006109f161138661190a565b84610aa3856040518060600160405280602581526020
001612b706025913960056006113b061190a565b6001600160a01b0390811682526020808301939093526040918 2
016000908120918d168152925290205491906 11c40565b60015460016001600160a01b0316331461142a5760405162461
bcd60e51b815260040180806020018281038252602381526020018061 2b4d60239139604001915050604051809103
90fd5b600254421161148057604080516 2461bcd60e51b815260206004820152601f60248201527f436f6e7472616
374206973206c6f636b656420756e74696c20372064617973006044820152905190819003606401 90fd5b60015460
0080546040516001600160a01b0393841693909116917f8be0079c531659141344cd1fd0a4f28419497f9722a3daa
fe3b4186f6b6457e091a3600154600080547 3ffffffffffffffffffffffffffffffffffffffff19166001600160a0
1b039092169190911790556 5b60006109f16114fb61190a565b84846119fa565b60025490565b61151061190a565
b600054600160016001600160a01b0390811691161461157257604080516 2461bcd60e51b815260206004820181905260248
01527f4f776e61626c653a2063616c6c6572206973206e6f7420746865206f776e657260448201529051908190036
0640190fd5b60138054821515610100810 27fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
ffff00ff909216919091179091556040805191825251 7f53726dfcaf90650aa7eb35524f4d3220f07413c8d6cb404
cc8c18bf5591bc1599181900360200190a150565b6115e661190a565b60005460016001600160a01b0390811691161461
164857604080516 2461bcd60e51b815260206004820181905260248201527f4f776e61626c653a2063616c6c65722
06973206e6f7420746865206f776e657260448201529051908190036064019 0fd5b611668606461166283600954 61

1e2e90919063ffffffff16565b90611cfa565b60145550565b61167661190a565b600054600160 0160a01b0390811
69116146116d8576040805162461bcd60e51b815260206004820181905260248201527f4f776e61626c653a206361
6c6c6572206973206e6f7420746865206f776e6572604482015290519081900360640190fd5b600080546001805473ffffffffffffffffffffffffffffffffffffffff199081166001600160a01b038416179091551681554282016002
5560405181907f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e0908290a350565b6
001600160a01b0391821660009081526005602090815260408083209390941682529190915220549056 5b61177061
190a565b600054600160 0160a01b0390811691161461 17d2576040805162461bcd60e51b815260206004820181905
260248201527f4f776e61626c653a2063616c6c657220697320 6e6f7420746865206f776e6572604482015290519 0
81900360640190fd5b6001600160a01b031660009081526006602052604090208054 60ff19169055565b6117fb611
90a565b600054600160 0160a01b0390811691161461185d576040805162461bcd60e51b815260206004820181905 2
60248201527f4f776e61626c653a2063616c6c657220697320 6e6f7420746865206f776e6572604482015290519 08
1900360640190fd5b6001600160a01b0381166118a2576040516 2461bcd60e51b8152600401808060200182810382
526026815260200180 6129f660269139604001915050604051809103 90fd5b600080546040516001600160a01b038
08516939216917f8be0079c531659141344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e091a36000805473
ffffffffffffffffffffffffffffffffffffffff19166001600160a01b0392909216919091179055565b3390565b6
001600160a01b038316611953576040516 2461bcd60e51b81526004018080602001828103825260248152602001 80
612afd6024913960400191505060405180910390fd5b6001600160a01b038216611998576040516 2461bcd60e51b8
15260040180806020018281038252602281526020018061 2a1c6022913960400191505060405180910390fd5b6001
600160a01b0380841660008181526005602090815260408083209487168084529482529182902085905581518581 5
291517f8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925928190039091 0190a35050
50565b6001600160a01b038316611a3f576040516 2461bcd60e51b815260040180806020018281038252602581526
0200180612ad86025913960400191505060405180910390fd5b6001600160a01b038216611a84576040516 2461bcd
60e51b815260040180806020018281038252602381526020018061 29a96023913960400191505060405180910390f
d5b60008111611ac3576040516 2461bcd60e51b815260040180806020018281038252602981526020018061 2aaf60
2991396040019150506040518091 0390fd5b611acb61129a565b6001600160a01b0316836001600160a01b0316141
58015611b055750611aef61129a565b6001600160a01b0316826001600160a01b031614155b15611b4b5760145481
1115611b4b576040516 2461bcd60e51b81526004018080602001828103825260288152602001806 12a3e602891396
04001915050604051809103 90fd5b6000611b5630611153565b9050601454811061 1b6657506014545b6015548110
8015908190611b7d575060135460ff16155b8015611bbb57507f00000000000000000000000000000000000000000
000000000000000000000006001600160a01b0316856001600160a01b031614155b8015611bce5750601354610100
900460ff165b15611be157601554915061 1be182611e87565b6001600160a01b038516600090815260066020526 04
0902054600190 60ff1680611c2357506001600160a01b038516600090815260066020526040902054 60ff165b1561
1c2c575060005b611c3886868684611f24565b505050505050565b60008184841115611ccf576040516 2461bcd60e
51b81526004018080602001828103825283818151815260200191508051906020019080838360005b83811015611c
9457818101518382015260200161 1c7c565b50505050905090810190601f168015611cc1578082038051600183602
0036101000a031916815260200191505b509250505060405180910390fd5b505050900390565b60008060006 11ce4
612098565b9092509050611cf38282611cfa565b9250505090565b6000611d3c838360405180604001604052806 01
a81526020017f536166654d6174683a206469766973696f6e206279207a65726f000000000000008152506121fb565b
9392505050565b6000828201838110156 11d3c576040805162461bcd60e51b815260206004820152601b602482015
27f536166654d6174683a206164646974696f6e206f766572666c6f770000000000604482015290519081900360 64
0190fd5b60008060008060008060008060006 11db48a612260565b925092509250600080600061 1dd28d8686611dc
d611cd7565b6122a2565b919f909e50909c50959a509398509196509294505050505050565b6000611d3c838360405 1
806040016040528060 1e81526020017f536166654d6174683a20737562747261637469 6f6e206f766572666c6f770
00081525 0611c40565b600082611e3d575060006109f5565b828202828482816 11e4a57fe5b04146 11d3c57604051
62461bcd60e51b815260040180806020018281038252602181526020018061 2a666021913960400191505060405 18
0910390fd5b6013805460ff1916600117905560006 11ea1826002611cfa565b90506000611eaf8383611dec565b90
5047611ebb836122f2565b6000611ec74783611dec565b9050611ed38382612501565b604080518581526020810 18
39052808201859052905 17f17bbfb9a6069321b6ded73bd96327c9e6b7212a5cd51ff219cd61370acafb561918190
03606001 90a150506013805460ff1916905550505050 565b80611f3157611f316125ff565b6001600160a01b0384166
0009081526007602052604090205460ff168015611f725750600160 0160a01b0383166000908152600760205260 40

90205460ff16155b15611f8757611f82848484612631565b612085565b6001600160a01b038416600090815260076
020526040902054 60ff16158015611fc85750600160016 0a01b038316600090815260076020526040902054 60ff16
5b15611fd857611f82848484612755565b6001600160a01b038416600090815260076020526040902054 60ff16158
01561201a57506001600160a01b038316600090815260076020526040902054 60ff16155b1561202a57611f828484
846127fe565b6001600160a01b038416600090815260076020526040902054 60ff16801561206a57506001600160a
01b038316600090815260076020526040902054 60ff165b1561207a57611f82848484612842565b61208584848461
27fe565b806120925761209261 28b5565b50505050565b600a546009546000918291825b6008548110156121c9578
260036000600884815481106120c157fe5b6000918252602080832090910154600160016 0a01b0316835282019290
925260400190205411806121265750816004600600884815481106120ff57fe5b6000918252602080832090910154
6001600160a01b0316835282019290925260400190205411 5b1561213d57600a546009549450945050506121f7
565b61217d6003600060088481548110612151 57fe5b6000918252602080832090910154600160016 0a01b0316835
282019290925260400190205484906 11dec565b92506121bf6004600060088481548110612193 57fe5b6000918252
602080832090910154600160016 0a01b0316835282019290925260400190205483906 11dec565b91506001016120a
5565b506009546 00a546121d991611cfa565b8210156121f157600a546009549350935050506121f7565b90925090
505b9091565b6000818361224a5760405162461bcd60e51b8152602060048201818152835160248401528351909 28
392604490910191908501908083836000831 5611c945781810151838201526020016 11c7c565b5060008385816122
5657fe5b0495945050505050565b6000806000806122 6f856128c3565b9050600061227c866128df565b905060006
122948261228e8986611dec565b90611dec565b979296509094509092505050565b60008080806122b18886611e2e
565b905060006122bf8887611e2e565b905060006122cd8888611e2e565b905060006122df8261228e8686611dec5
65b939b939a509198509196505050505050565b604080516002808252606080830184529260208301908036833 7
019050509050308160008151811061232057fe5b6020026020010190600160016 0a01b0316908160016 00a01b0
316815250507f0000000000000000000000000000000000000000000000000000000000000000000000006001600160a01b03
1663ad5c46486040518163ffffffff1660e01b8152600401602060405180830381868 03b1580156123995760008 0f
d5b505afa1580156123ad573d6000803e3d6000fd5b50505050640513d60208110156123c357600080fd5b505181
5182906001908110612 3d457fe5b6020026020010190600160016 0a01b0316908160016 00a01b0316815250506
1241f307f0000000000000000000000000000000000000000000000000000000008461190e565b7f000000
0000000000000000000000000000000000000000000000000000000000006001600160a01b031663791ac9478360008
430426040518663ffffffff1660e01b8152600401808681526020018581526020018060200184600160016 0a01b03
168152602001838152602001828103825285818151815260200191508051906020019060200280838360005b83811
0156124c45781810151838201526020016124ac565b505050509050019650505050505050600060405180830381 60
0087803b1580156124ed57600080fd5b505af1158015611c38573d6000803e3d6000fd5b61252c307f00000000000
0000000000000000000000000000000000000000000008461190e565b7f000000000000000000000000000
0000000000000000000000000000000006001600160a01b031663f305d719823085600080612569611 29a565
b426040518863ffffffff1660e01b8152600401808760016001600160a01b0316815260200186815260200185815260 20
018481526020018360016001600160a01b0316815260200182815260200196505050505050506060604051808303818 58
8803b1580156125d457600080fd5b505af11580156125e8573d6000803e3d6000fd5b505050506040513d606081
101561209257600080fd5b600f5415801561260f5750601154155b156126195761262f565b600f805460105560118
0546012556000918290555555b565b600080600080600080612643876 11d9d565b6001600160a01b038f1660009081
526004602052604090205495 9b5093995091975095509350915061267590886 11dec565b6001600160a01b038a166
0009081526004602090815260408083209390935560039052205 46126a49087611dec565b6001600160a01b03808b
166000908152600360205260408082209390935590 8a16815220546126d39086611d43565b6001600160a01b03891
660009081526003602052604090205561 26f5816128fb565b6126ff8483612984565b876001600160a01b03168960
01600160a01b03167fddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef85604051808
28152602001915050604051809103 90a3505050505050505050565b60008060008060008061276787611d9d565b60
01600160a01b038f166000908152600360205260409020 54959b5093995091975095509350915061279990876 11de
c565b6001600160a01b03808b166000908152600360209081526040808320949094559 18b16815260049091522054
6127cf9084611d43565b6001600160a01b038916600090815260046020908152604080832093909355600390520 5
46126d39086611d43565b60008060008060008061281087611d9d565b6001600160a01b038f166000908152600360
205260409020 54959b5093995091975095509350915061506126a49087611dec565b60008060008060008061285487611
d9d565b6001600160a01b038f166000908152600460205260409020 54959b5093995091975095509350915061288 6

```
9088611dec565b6001600160a01b038a16600090815260046020908152604080832093909355600390522054612799087611dec565b601054600f5560125460115556 5b60006109f56064611662600f5485611e2e90919063ffffffff16565b60006109f56064611662601154 85611e2e90919063ffffffff16565b6000612905611cd7565b90506000612138383611e2e565b30600090815260036020526040902054909150612930908261 1d43565b306000908152600360205260409020555b505050565b600a546129919083611dec565b600a55600b546129a19082611d43565b600b55505056fe45524332303a207472616e7366657220746f20746865207a65726f2061646472657373735472616e7366657220616d6f756e7420657863656564732074686520616d6f756e742e536166654d6174683a206d756c7469706c69636174696f6e206f766572666c6f7745524332303a207472616e7366657220616d6f756e74206578636565647320616c6c6f77616e636572616e7366657220616d6f756e74206d757374206265206c657373207468616e20746f74616c207265666c656374696f6e734f776e61626c653a206e6577206f776e657220697320746865207a65726f2061646472657373345524332303a20617070726f766520746f20746865207a65726f2061646472657373472616e7366657220616d6f756e7420657863616e6f7420617070726f766520746f20746865207a65726f2061646472657373578636c7564656420616464726573736573206361 6e6e6f742063616c6c207468697320066756e6374696f6e596f7520646f6e27742068617665207065726d697373696f6e2074 6f20756e6c6f636b45524332303a206465637265617365420616c6c6f77616e63652062656c6f77207a65726fa26 4697066735822122091d31f7aefafff2a2ca60aa572b0e3bfb18959ee7e58d2ab4e6b582624e9624664736f6c634300060c0033"
```