```
1 // SPDX-License-Identifier: AGPL-3.0-or-later
 2 pragma solidity 0.7.5;
 4 interface IERC20 {
 5
      * @dev Returns the amount of tokens in existenc
   е.
 7
    function totalSupply() external view returns (uin
   t256);
10
      * @dev Returns the amount of tokens owned by `ac
   count`.
12
    function balanceOf(address account) external view
13
   returns (uint256);
14
15
      * @dev Moves `amount` tokens from the caller's a
   ccount to `recipient`.
17
      * Returns a boolean value indicating whether the
18
   operation succeeded.
19
20
      * Emits a {Transfer} event.
22
    function transfer(address recipient, uint256 amou
   nt) external returns (bool);
24
      ^{\ast} @dev Returns the remaining number of tokens th
25
   at `spender` will be
       * allowed to spend on behalf of `owner` through
    {transferFrom}. This is
27
      * zero by default.
28
      ^{\star} This value changes when {approve} or {transfer \,
29
   From} are called.
30
    function allowance(address owner, address spende
   r) external view returns (uint256);
32
33
      * @dev Sets `amount` as the allowance of `spende
34
   r` over the caller's tokens.
35
      * Returns a boolean value indicating whether the
   operation succeeded.
      * IMPORTANT: Beware that changing an allowance w
38
   ith this method brings the risk
      ^{\ast} 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 all
   owance to 0 and set the
      * desired value afterwards:
42
      * https://github.com/ethereum/EIPs/issues/20#iss
43
   uecomment - 263524729
```

* Emits an {Approval} event.

```
1 // SPDX-License-Identifier: AGPL-3.0-or-later
 2 pragma solidity 0.7.5;
4 interface IERC20 {
 5
     * @dev Returns the amount of tokens in existenc
   е.
7
8 function totalSupply() external view returns (uin
   t256);
10
     * @dev Returns the amount of tokens owned by `ac
   count`.
12
   function balanceOf(address account) external view
13
   returns (uint256);
14
15
     * @dev Moves `amount` tokens from the caller's a
   ccount to `recipient`.
17
      * Returns a boolean value indicating whether the
18
   operation succeeded.
19
20
      * Emits a {Transfer} event.
    function transfer(address recipient, uint256 amou
   nt) external returns (bool);
24
     ^{\ast} @dev Returns the remaining number of tokens th
25
   at `spender` will be
       * allowed to spend on behalf of `owner` through
    {transferFrom}. This is
      * zero by default.
27
28
      * This value changes when {approve} or {transfer
29
   From} are called.
30
   function allowance(address owner, address spende
   r) external view returns (uint256);
32
33
     * @dev Sets `amount` as the allowance of `spende
34
   r` over the caller's tokens.
35
     * Returns a boolean value indicating whether the
   operation succeeded.
     * IMPORTANT: Beware that changing an allowance w
   ith 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 all
   owance to 0 and set the
      * desired value afterwards:
42
      * https://github.com/ethereum/EIPs/issues/20#iss
43
   uecomment-263524729
44
      * Emits an {Approval} event.
```

```
function approve(address spender, uint256 amount)
     function approve(address spender, uint256 amount)
   external returns (bool);
                                                                external returns (bool);
49
                                                             49
     * @dev Moves `amount` tokens from `sender` to `r
                                                                  * @dev Moves `amount` tokens from `sender` to `r
50
                                                             50
   ecipient` using the
                                                                ecipient` using the
     * allowance mechanism. `amount` is then deducted
                                                                   * allowance mechanism. `amount` is then deducted
   from the caller's
                                                                from the caller's
      * allowance.
                                                                   * allowance.
                                                             52
52
53
                                                             53
      ^{\star} Returns a boolean value indicating whether the
                                                                   ^{\star} Returns a boolean value indicating whether the
   operation succeeded.
                                                                 operation succeeded.
                                                             55
      * Emits a {Transfer} event.
                                                                   * Emits a {Transfer} event.
56
                                                             56
57
                                                             57
    function transferFrom(address sender, address rec
                                                                 function transferFrom(address sender, address rec
58
                                                             58
   ipient, uint256 amount) external returns (bool):
                                                                ipient, uint256 amount) external returns (bool);
60
                                                             60
     * @dev Emitted when `value` tokens are moved fro
                                                                   * @dev Emitted when `value` tokens are moved fro
61
                                                             61
   m one account (`from`) to
                                                                m one account (`from`) to
62
     * another (`to`).
                                                             62
                                                                   * another (`to`).
63
                                                             63
      * Note that `value` may be zero.
                                                                   * Note that `value` may be zero.
64
                                                             64
65
                                                             65
    event Transfer(address indexed from, address inde
                                                                 event Transfer(address indexed from, address inde
   xed to, uint256 value);
                                                                 xed to, uint256 value);
                                                             67
67
68
                                                             68 /**
    * @dev Emitted when the allowance of a `spender`
                                                             69
                                                                 * @dev Emitted when the allowance of a `spender`
   for an `owner` is set by
                                                                for an `owner` is set by
    * a call to {approve}. `value` is the new allowa
                                                                 * a call to {approve}. `value` is the new allowa
70
   nce.
                                                                nce.
71
     event Approval(address indexed owner, address ind
                                                                  event Approval(address indexed owner, address ind
   exed spender, uint256 value);
                                                                 exed spender, uint256 value);
73 }
                                                             73 }
74
                                                             74
75 library LowGasSafeMath {
                                                             75 library LowGasSafeMath {
     /// @notice Returns x + y, reverts if sum overf
                                                                 /// @notice Returns x + y, reverts if sum overf
   lows uint256
                                                                 lows uint256
77
     /// @param x The augend
                                                                  /// @param x The augend
       /// @param y The addend
                                                                    /// @param y The addend
       /// @return z The sum of x and y
                                                             79
                                                                    /// @return z The sum of x and y
       function add(uint256 x, uint256 y) internal pur
                                                                    function add(uint256 x, uint256 y) internal pur
                                                             80
   e returns (uint256 z) {
                                                                e returns (uint256 z) {
81
           require((z = x + y) >= x);
                                                             81
                                                                        require((z = x + y) >= x);
82
                                                             82
83
                                                             83
       /// @notice Returns x - y, reverts if underflow
                                                                    /// @notice Returns x - y, reverts if underflow
85
       /// @param x The minuend
                                                             85
                                                                    /// @param x The minuend
       /// @param y The subtrahend
86
                                                             86
                                                                    /// @param y The subtrahend
87
       /// @return z The difference of x and y
                                                             87
                                                                    /// @return z The difference of \boldsymbol{x} and \boldsymbol{y}
88
       function sub(uint256 x, uint256 y) internal pur
                                                             88
                                                                    function sub(uint256 x, uint256 y) internal pur
   e returns (uint256 z) {
                                                                 e returns (uint256 z) {
89
           require((z = x - y) <= x);
                                                             89
                                                                        require((z = x - y) <= x);
91
                                                             91
       /// @notice Returns x ^{\ast} y, reverts if overflows
                                                                    /// @notice Returns x ^{\ast} y, reverts if overflows
92
                                                             92
       /// @param x The multiplicand
                                                                    /// @param x The multiplicand
93
                                                             93
94
       /// @param y The multiplier
                                                             94
                                                                    /// @param y The multiplier
95
       /// @return z The product of x and y
                                                             95
                                                                    /// @return z The product of x and y
       function mul(uint256 x, uint256 y) internal pur
                                                                    function mul(uint256 x, uint256 y) internal pur
   e returns (uint256 z) {
                                                                e returns (uint256 z) {
           require(x == 0 || (z = x * y) / x == y);
                                                             97
                                                                        require(x == 0 || (z = x * y) / x == y);
98
                                                             98
                                                             99
```

46

```
100
        /// @notice Returns x + y, reverts if overflows
                                                           100
                                                                    /// @notice Returns x + y, reverts if overflows
    or underflows
                                                                or underflows
       /// @param x The augend
                                                                    /// @param x The augend
101
                                                            101
102
      /// @param y The addend
                                                            102
                                                                    /// @param y The addend
103
     /// @return z The sum of x and y
                                                            103
                                                                    /// @return z The sum of x and y
       function add(int256 x, int256 y) internal pure
                                                                  function add(int256 x, int256 y) internal pure
     returns (int256 z) {
                                                                 returns (int256 z) {
105
            require((z = x + y) >= x == (y >= 0));
                                                           105
                                                                        require((z = x + y) >= x == (y >= 0));
106
                                                            106
107
        /// @notice Returns x - y, reverts if overflows
                                                                    /// @notice Returns x - y, reverts if overflows
108
                                                            108
    or underflows
                                                              or underflows
      /// @param x The minuend
                                                            109
                                                                  /// @param x The minuend
109
      /// @param y The subtrahend
                                                            110 /// @param y The subtrahend
111
        /// @return z The difference of x and y
                                                            111
                                                                    /// @return z The difference of x and y
       function sub(int256 x, int256 y) internal pure
                                                                  function sub(int256 x, int256 y) internal pure
     returns (int256 z) {
                                                                 returns (int256 z) {
            require((z = x - y) \leq x == (y \geq 0));
                                                                        require((z = x - y) <= x == (y >= 0));
113
                                                           113
114
                                                            114
115 }
                                                            115 }
                                                            116
117 abstract contract ERC20 is IERC20 {
                                                            117 abstract contract ERC20 is IERC20 {
118
                                                            118
      using LowGasSafeMath for uint256;
                                                                  using LowGasSafeMath for uint256;
119
                                                            119
120
                                                            120
121
      // Present in FRC777
                                                            121
                                                                  // Present in ERC777
122
      mapping (address => uint256) internal _balances;
                                                            122
                                                                  mapping (address => uint256) internal _balances;
123
                                                            123
124
      // Present in ERC777
                                                            124
                                                                  // Present in ERC777
      mapping (address => mapping (address => uint256))
                                                                  mapping (address => mapping (address => uint256))
    internal _allowances;
                                                                internal _allowances;
126
                                                            126
      // Present in ERC777
                                                                  // Present in ERC777
                                                            127
127
128
      uint256 internal _totalSupply;
                                                            128
                                                                  uint256 internal _totalSupply;
129
                                                            129
130
      // Present in ERC777
                                                                 // Present in ERC777
                                                            130
      string internal _name;
                                                                  string internal _name;
131
132
                                                            132
133
      // Present in ERC777
                                                            133
                                                                  // Present in ERC777
     string internal _symbol;
                                                                  string internal _symbol;
134
                                                            134
135
                                                            135
136
      // Present in ERC777
                                                            136
                                                                  // Present in ERC777
137
      uint8 internal _decimals;
                                                            137
                                                                  uint8 internal _decimals;
138
                                                            138
     constructor (string memory name_, string memory s
                                                                 constructor (string memory name_, string memory s
    ymbol_, uint8 decimals_) {
                                                                ymbol , uint8 decimals ) {
        name = name ;
                                                            140
                                                                    name = name ;
140
        _symbol = symbol_;
                                                                    _symbol = symbol_;
141
                                                            141
                                                                    _decimals = decimals_;
        _decimals = decimals_;
142
                                                            142
143
                                                            143
144
                                                            144
      function name() public view returns (string memor
                                                                  function name() public view returns (string memor
       return _name;
                                                                    return _name;
146
                                                            146
                                                            147
147
148
149
      function symbol() public view returns (string mem
                                                            149
                                                                  function symbol() public view returns (string mem
    ory) {
                                                                ory) {
150
        return _symbol;
                                                            150
                                                                    return _symbol;
151
                                                            151
152
                                                            152
153
      function decimals() public view returns (uint8) {
                                                            153
                                                                  function decimals() public view returns (uint8) {
      return _decimals;
                                                                   return _decimals;
154
                                                            154
155
                                                            155
                                                            156
156
      function totalSupply() public view override retur
                                                                  function totalSupply() public view override retur
    ns (uint256) {
                                                                ns (uint256) {
158
        return _totalSupply;
                                                            158
                                                                    return _totalSupply;
                                                            159
159
      }
                                                                  }
                                                            160
160
```

```
function balanceOf(address account) public view v
                                                                    function balanceOf(address account) public view v
161
                                                              161
    irtual override returns (uint256) {
                                                                  irtual override returns (uint256) {
        return _balances[account];
                                                                      return _balances[account];
162
                                                              162
163
                                                              163
      }
                                                                    }
164
      function transfer(address recipient, uint256 amou
                                                                    function transfer(address recipient, uint256 amou
165
                                                              165
    nt) public virtual override returns (bool) {
                                                                  nt) public virtual override returns (bool) {
166
        _transfer(msg.sender, recipient, amount);
                                                              166
                                                                      _transfer(msg.sender, recipient, amount);
        return true;
                                                                      return true;
167
                                                              167
                                                              168
168
      }
                                                                    }
                                                              169
170
        function allowance(address owner, address spend
                                                              170
                                                                       function allowance(address owner, address spend
    er) public view virtual override returns (uint256)
                                                                  er) public view virtual override returns (uint256)
171
            return _allowances[owner][spender];
                                                              171
                                                                           return _allowances[owner][spender];
                                                              172
173
                                                              173
        function approve(address spender, uint256 amoun
                                                                       function approve(address spender, uint256 amoun
174
                                                              174
    t) public virtual override returns (bool) {
                                                                  t) public virtual override returns (bool) {
175
             _approve(msg.sender, spender, amount);
                                                              175
                                                                           _approve(msg.sender, spender, amount);
176
            return true;
                                                              176
                                                                           return true;
                                                              177
178
                                                              178
179
        function transferFrom(address sender, address r
                                                              179
                                                                      function transferFrom(address sender, address r
    ecipient, uint256 amount) public virtual override r
                                                                  ecipient, uint256 amount) public virtual override r
    eturns (bool) {
                                                                  eturns (bool) {
180
            _transfer(sender, recipient, amount);
                                                              180
                                                                           _transfer(sender, recipient, amount);
181
            _approve(sender, msg.sender, _allowances[se
                                                              181
                                                                           _approve(sender, msg.sender, _allowances[se
    nder][msq.sender]
                                                                  nder][msq.sender]
182
              .sub(amount));
                                                              182
                                                                            .sub(amount));
            return true;
                                                              183
                                                                           return true;
183
184
                                                              184
186
        function increaseAllowance(address spender, uin
                                                              186
                                                                      function increaseAllowance(address spender, uin
    t256 addedValue) public virtual returns (bool) {
                                                                  t256 addedValue) public virtual returns (bool) {
187
            _approve(msg.sender, spender, _allowances[m
                                                                           _approve(msg.sender, spender, _allowances[m
    sg.sender][spender].add(addedValue));
                                                                  sg.sender][spender].add(addedValue));
            return true;
                                                              188
                                                                          return true;
188
189
                                                              189
                                                              190
191
        function decreaseAllowance(address spender, uin
                                                              191
                                                                      function decreaseAllowance(address spender, uin
    t256 subtractedValue) public virtual returns (bool)
                                                                  t256 subtractedValue) public virtual returns (bool)
192
            _approve(msg.sender, spender, _allowances[m
                                                              192
                                                                           _approve(msg.sender, spender, _allowances[m
    sg.sender][spender]
                                                                  sg.sender][spender]
193
              .sub(subtractedValue));
                                                              193
                                                                            .sub(subtractedValue));
            return true:
                                                              194
                                                                           return true:
194
195
        }
                                                              195
                                                                      }
196
197
        function _transfer(address sender, address reci
                                                                      function _transfer(address sender, address reci
                                                              197
    pient, uint256 amount) internal virtual {
                                                                  pient, uint256 amount) internal virtual {
198
          require(sender != address(0), "ERC20: transfe
                                                                        require(sender != address(0), "ERC20: transfe
    r from the zero address");
                                                                  r from the zero address");
          require(recipient != address(0), "ERC20: tran
                                                              199
                                                                        require(recipient != address(0), "ERC20: tran
199
    sfer to the zero address");
                                                                  sfer to the zero address");
200
201
          _beforeTokenTransfer(sender, recipient, amoun
                                                              201
                                                                         _beforeTokenTransfer(sender, recipient, amoun
    t);
                                                                  t);
                                                              202
202
203
          _balances[sender] = _balances[sender].sub(amo
                                                              203
                                                                         _balances[sender] = _balances[sender].sub(amo
    unt):
                                                                  unt);
204
                                                              204
          _balances[recipient] = _balances[recipient].a
                                                                         _balances[recipient] = _balances[recipient].a
    dd(amount);
                                                                  dd(amount);
205
          emit Transfer(sender, recipient, amount);
                                                              205
                                                                         emit Transfer(sender, recipient, amount);
206
                                                              206
207
                                                              207
        function _mint(address account_, uint256 amount
                                                                       function _mint(address account_, uint256 amount
     _) internal virtual {
                                                                  _) internal virtual {
            require(account != address(0), "ERC20: min
                                                              209
                                                                          require(account != address(0), "ERC20: min
    t to the zero address");
                                                                  t to the zero address");
```

```
_beforeTokenTransfer(address( this ), accou
                                                            210
                                                                         _beforeTokenTransfer(address( this ), accou
    nt_, amount );
                                                                nt_, amount_);
            _totalSupply = _totalSupply.add(amount_);
                                                                         _totalSupply = _totalSupply.add(amount_);
211
                                                             211
            _balances[account_] = _balances[account_].a
                                                                         _balances[account_] = _balances[account_].a
    dd(amount_);
                                                                 dd(amount_);
            emit Transfer(address(0), account_, amount
                                                                         emit Transfer(address(0), account_, amount
                                                             213
    _);
                                                                 _);
214
                                                             214
215
                                                             215
        function burn(address account, uint256 amount)
                                                                     function burn(address account, uint256 amount)
                                                             216
216
    internal virtual {
                                                                 internal virtual {
                                                                       require(account != address(0), "ERC20: burn
217
           require(account != address(0), "ERC20: burn
                                                             217
    from the zero address");
                                                                 from the zero address");
                                                             218
218
219
            _beforeTokenTransfer(account, address(0), a
                                                                         _beforeTokenTransfer(account, address(0), a
                                                                 mount);
    mount);
220
                                                             220
            _balances[account] = _balances[account].sub
                                                             221
                                                                         _balances[account] = _balances[account].sub
221
    (amount);
                                                                 (amount);
222
            _totalSupply = _totalSupply.sub(amount);
                                                             222
                                                                         _totalSupply = _totalSupply.sub(amount);
            emit Transfer(account, address(0), amount);
                                                             223
                                                                          emit Transfer(account, address(0), amount);
224
                                                             224
225
                                                             225
226
        function approve(address owner, address spende
                                                             226
                                                                     function approve(address owner, address spende
    r, uint256 amount) internal virtual {
                                                                 r, uint256 amount) internal virtual {
            require(owner != address(0), "ERC20: approv
                                                                         require(owner != address(0), "ERC20: approv
227
    e from the zero address");
                                                                 e from the zero address");
            require(spender != address(0), "ERC20: appr
                                                                         require(spender != address(0), "ERC20: appr
228
                                                             228
    ove to the zero address");
                                                                 ove to the zero address");
                                                             229
            _allowances[owner][spender] = amount;
                                                                         _allowances[owner][spender] = amount;
230
                                                             230
            emit Approval(owner, spender, amount);
231
                                                             231
                                                                         emit Approval(owner, spender, amount);
232
                                                             232
233
                                                             233
234
      function _beforeTokenTransfer( address from_, add
                                                             234
                                                                   function _beforeTokenTransfer( address from_, add
    ress to_, uint256 amount_ ) internal virtual { }
                                                                  ress to_, uint256 amount_ ) internal virtual { }
235 }
                                                             235 }
236
                                                             236
237 library Counters {
                                                             237 library Counters {
        using LowGasSafeMath for uint256;
                                                                     using LowGasSafeMath for uint256;
238
                                                             238
239
                                                             239
240
        struct Counter {
                                                             240
                                                                     struct Counter {
            uint256 _value; // default: 0
                                                                         uint256 _value; // default: 0
241
                                                             241
242
                                                             242
243
                                                             243
        function current(Counter storage counter) inter
                                                                     function current(Counter storage counter) inter
    nal view returns (uint256) {
                                                                 nal view returns (uint256) {
245
            return counter._value;
                                                             245
                                                                         return counter, value:
                                                             246
246
247
                                                             247
        function increment(Counter storage counter) int
                                                                     function increment(Counter storage counter) int
249
            counter._value += 1;
                                                             249
                                                                         counter._value += 1;
250
                                                             250
                                                             251
        function decrement(Counter storage counter) int
                                                             252
                                                                     function decrement(Counter storage counter) int
252
    ernal {
                                                                 ernal {
253
            counter._value = counter._value.sub(1);
                                                             253
                                                                         counter._value = counter._value.sub(1);
                                                             254
254
                                                             255 }
256
                                                             256
257 interface IERC2612Permit {
                                                             257 interface IERC2612Permit {
                                                             258
258
259
        function permit(
                                                             259
                                                                     function permit(
260
            address owner,
                                                             260
                                                                         address owner,
261
            address spender,
                                                             261
                                                                         address spender,
            uint256 amount,
                                                                         uint256 amount,
            uint256 deadline,
                                                                         uint256 deadline,
264
            uint8 v,
                                                             264
                                                                         uint8 v,
265
            bvtes32 r,
                                                             265
                                                                         bvtes32 r,
```

```
267
                                                                267
          ) external:
                                                                         ) external:
                                                                 268
          function nonces(address owner) external view re
                                                                         function nonces(address owner) external view re
      turns (uint256);
                                                                     turns (uint256);
 270
                                                                270 }
 271
                                                                271
 272 abstract contract ERC20Permit is ERC20, IERC2612Per
                                                                272 abstract contract ERC20Permit is ERC20, IERC2612Per
 273
          using Counters for Counters. Counter:
                                                                273
                                                                         using Counters for Counters. Counter:
 274
                                                                274
          mapping(address => Counters.Counter) private _n
                                                                         mapping(address => Counters.Counter) private _n
      onces;
                                                                     onces;
 276
                                                                276
          // keccak256("Permit(address owner,address spen
                                                                         // keccak256("Permit(address owner,address spen
      der, uint256 value, uint256 nonce, uint256 deadlin
                                                                     der, uint256 value, uint256 nonce, uint256 deadlin
      e)");
                                                                     e)");
 278
          bytes32 public constant PERMIT TYPEHASH = 0x6e7
                                                                278
                                                                         bytes32 public constant PERMIT TYPEHASH = 0x6e7
      1edae12b1b97f4d1f60370fef10105fa2faae0126114a169c64
                                                                     1edae12b1b97f4d1f60370fef10105fa2faae0126114a169c64
      845d6126c9;
                                                                     845d6126c9;
 279
                                                                279
          bytes32 public DOMAIN_SEPARATOR;
                                                                         bytes32 public DOMAIN_SEPARATOR;
 280
 281
                                                                281
 282
          constructor() {
                                                                282
                                                                         constructor() {
 283
              uint256 chainID;
                                                                283
                                                                             uint256 chainID;
 284
              assembly {
                                                                284
                                                                             assembly {
                  chainID := chainid()
                                                                                  chainID := chainid()
 285
                                                                285
 286
                                                                286
                                                                287
 287
              DOMAIN_SEPARATOR = keccak256(
                                                                             DOMAIN_SEPARATOR = keccak256(
 288
                                                                288
 289
                  abi.encode(
                                                                                  abi.encode(
 290
                       keccak256("EIP712Domain(string nam
                                                                                      keccak256("EIP712Domain(string nam
      e, string version, uint256 chainId, address verifyingC
                                                                     e, string version, uint256 chainId, address verifyingC
      ontract)"),
                                                                     ontract)"),
291
                       keccak256(bytes(name())),
                                                                291
                                                                                      keccak256(bytes(name())),
 292
                       keccak256(bytes("1")), // Version
                                                                292
                                                                                      keccak256(bytes("1")), // Version
 293
                       chainID,
                                                                                      chainID,
                                                                 293
                       address(this)
 294
                                                                 294
                                                                                      address(this)
 295
                                                                 295
                                                                296
 296
              );
                                                                             );
 297
          }
                                                                297
                                                                         }
 298
                                                                 298
 299
          function permit(
                                                                299
                                                                         function permit(
              address owner,
                                                                 300
                                                                             address owner,
 300
              address spender,
                                                                             address spender,
 301
                                                                301
              uint256 amount.
                                                                             uint256 amount,
              uint256 deadline,
                                                                303
                                                                             uint256 deadline,
 303
              uint8 v.
                                                                304
                                                                             uint8 v.
 304
                                                                             bytes32 r,
 305
              bytes32 r,
                                                                305
 306
              bytes32 s
                                                                306
                                                                             bytes32 s
 307
          ) public virtual override {
                                                                307
                                                                         ) public virtual override {
              require(block.timestamp <= deadline, "Permi</pre>
                                                                             require(block.timestamp <= deadline, "Permi</pre>
      t: expired deadline");
                                                                     t: expired deadline");
 309
                                                                309
 310
              bvtes32 hashStruct =
                                                                310
                                                                             bvtes32 hashStruct =
 311
                  keccak256(abi.encode(PERMIT TYPEHASH, o
                                                                311
                                                                                  keccak256(abi.encode(PERMIT TYPEHASH, o
      wner, spender, amount, _nonces[owner].current(), de
                                                                     wner, spender, amount, _nonces[owner].current(), de
      adline));
                                                                     adline));
 312
                                                                312
 313
              bytes32 _hash = keccak256(abi.encodePacked
                                                                             bytes32 _hash = keccak256(abi.encodePacked
      (uint16(0x1901), DOMAIN_SEPARATOR, hashStruct));
                                                                     (uint16(0x1901), DOMAIN_SEPARATOR, hashStruct));
 314
                                                                314
 315
              address signer = ecrecover(_hash, v, r, s);
                                                                             address signer = ecrecover( hash, v, r, s);
              require(signer != address(0) && signer == o
                                                                             require(signer != address(0) && signer == o
 316
                                                                316
      wner, "ERC20Permit: Invalid signature");
                                                                     wner, "ERC20Permit: Invalid signature");
 317
                                                                317
              _nonces[owner].increment();
 318
                                                                318
                                                                             _nonces[owner].increment();
 319
              _approve(owner, spender, amount);
                                                                319
                                                                             _approve(owner, spender, amount);
 320
                                                                320
          }
                                                                         }
                                                                321
 321
```

bytes32 s

266

bytes32 s

```
function nonces(address owner) public view over
                                                            322
                                                                     function nonces(address owner) public view over
     ride returns (uint256) {
                                                                 ride returns (uint256) {
                                                                         return _nonces[owner].current();
           return _nonces[owner].current();
 323
                                                             323
 324
                                                             324
 325 }
                                                             325 }
 326
                                                             326
 327 interface IOwnable {
                                                             327 interface IOwnable {
       function owner() external view returns (address);
                                                             328
                                                                    function owner() external view returns (address);
 329
                                                             329
      function renounceOwnership() external:
                                                                   function renounceOwnership() external:
 330
                                                             330
 332
       function transferOwnership( address newOwner_ ) e
                                                             332
                                                                   function transferOwnership( address newOwner_ ) e
      xternal;
                                                                  xternal;
 333 }
                                                             333 }
                                                             334
 334
 335 contract Ownable is IOwnable {
                                                             335 contract Ownable is IOwnable {
 336
                                                             336
 337
       address internal owner:
                                                             337
                                                                    address internal owner:
 338
 339
       event OwnershipTransferred(address indexed previo
                                                             339
                                                                   event OwnershipTransferred(address indexed previo
     usOwner, address indexed newOwner);
                                                                  usOwner, address indexed newOwner);
                                                             340
 340
 341
       constructor () {
                                                             341
                                                                   constructor () {
 342
                                                             342
                                                                     _owner = msg.sender;
         owner = msq.sender;
 343
         emit OwnershipTransferred( address(0), _owner
                                                             343
                                                                     emit OwnershipTransferred( address(0), _owner
      );
                                                                   );
 344
       }
                                                             344
                                                                   }
 345
                                                             345
       function owner() public view override returns (ad
                                                                   function owner() public view override returns (ad
     dress) {
                                                                  dress) {
 347
       return _owner;
                                                             347
                                                                    return _owner;
 348
                                                             348
       }
 349
                                                             349
 350
       modifier onlyOwner() {
                                                             350
                                                                    modifier onlyOwner() {
351
       require( _owner == msg.sender, "Ownable: caller
                                                                    require( _owner == msg.sender, "Ownable: caller
     is not the owner" );
                                                                  is not the owner" );
                                                             352
 352
         _;
                                                                     _;
 353
                                                             353
 354
                                                             354
                                                                   function renounceOwnership() public virtual overr
 355
       function renounceOwnership() public virtual overr
                                                             355
     ide onlvOwner() {
                                                                  ide onlvOwner() {
         emit OwnershipTransferred( _owner, address(0)
                                                                     emit OwnershipTransferred( _owner, address(0)
 356
                                                             356
 357
         _owner = address(0);
                                                             357
                                                                     _owner = address(0);
 358
                                                             358
 359
                                                             359
      function transferOwnership( address newOwner_ ) p
                                                             360 function transferOwnership( address newOwner_ ) p
 360
     ublic virtual override onlyOwner() {
                                                                  ublic virtual override onlyOwner() {
        require( newOwner_ != address(0), "Ownable: new
                                                                  require( newOwner_ != address(0), "Ownable: new
 361
                                                             361
     owner is the zero address");
                                                                  owner is the zero address");
         emit OwnershipTransferred( _owner, newOwner_ );
                                                                      emit OwnershipTransferred( _owner, newOwner_ );
         owner = newOwner ;
                                                             363
                                                                      owner = newOwner ;
 364
                                                             364
       }
                                                             365 }
 365 }
                                                             366
 367 contract VaultOwned is Ownable {
                                                             367 contract VaultOwned is Ownable {
 368
                                                             368
 369
       address internal _vault;
                                                             369
                                                                    address internal _vault;
 370
                                                             370
       event VaultTransferred(address indexed newVault);
                                                                    event VaultTransferred(address indexed newVault);
 371
                                                             371
 372
                                                             372
 373
       function setVault( address vault_ ) external only
                                                             373
                                                                    function setVault( address vault_ ) external only
     Owner() {
                                                                  Owner() {
       require(vault_ != address(0), "IAO");
                                                             374
                                                                    require(vault != address(0), "IAO");
 374
 375
         _vault = vault_;
                                                             375
                                                                     _vault = vault_;
 376
         emit VaultTransferred( _vault );
                                                             376
                                                                      emit VaultTransferred( _vault );
 377
                                                             377
 378
                                                             378
 379
       function vault() public view returns (address) {
                                                             379
                                                                   function vault() public view returns (address) {
 380
         return vault:
                                                             380
                                                                     return vault:
```

```
381
                                                             381
      }
                                                                   }
382
                                                             382
383
      modifier onlyVault() {
                                                                  modifier onlyVault() {
                                                             383
        require( _vault == msg.sender, "VaultOwned: cal
                                                                     require( _vault == msg.sender, "VaultOwned: cal
384
                                                             384
    ler is not the Vault" );
                                                                  ler is not the Vault" );
385
                                                             385
                                                             386
      }
387
                                                             387
388
                                                             388
389
                                                             389
    contract TimeERC20Token is ERC20Permit, VaultOwned
                                                                 contract MaiaERC20Token is ERC20Permit, VaultOwned
390
                                                             390
                                                             391
391
392
        using LowGasSafeMath for uint256;
                                                             392
                                                                      using LowGasSafeMath for uint256;
                                                             393
393
        constructor() ERC20("Time", "TIME", 9) {
                                                                     constructor() ERC20("Maia", "MAIA", 9) {
394
                                                             394
395
        }
                                                             395
396
                                                             396
        function mint(address account_, uint256 amount
                                                             397
                                                                      function mint(address account_, uint256 amount
397
    _) external onlyVault() {
                                                                  _) external onlyVault() {
398
            _mint(account_, amount_);
                                                             398
                                                                         _mint(account_, amount_);
399
                                                             399
                                                             400
400
401
        function burn(uint256 amount) external virtual
                                                             401
                                                                      function burn(uint256 amount) external virtual
     {
                                                                  {
402
                                                             402
            burn(msg.sender, amount);
                                                                          burn(msq.sender, amount);
403
                                                             403
        }
                                                                     }
404
                                                             404
405
        function burnFrom(address account_, uint256 amo
                                                             405
                                                                     function burnFrom(address account_, uint256 amo
    unt_) external virtual {
                                                                 unt_) external virtual {
            _burnFrom(account_, amount_);
                                                                         _burnFrom(account_, amount_);
407
                                                             407
408
                                                             408
        function _burnFrom(address account_, uint256 am
                                                                     function _burnFrom(address account_, uint256 am
409
                                                             409
                                                                 ount_) internal virtual {
    ount_) internal virtual {
410
           uint256 decreasedAllowance_ =
                                                             410
                                                                         uint256 decreasedAllowance_ =
                allowance(account_, msg.sender).sub(amo
                                                                             allowance(account_, msg.sender).sub(amo
    unt_);
                                                                 unt_);
412
                                                             412
                                                                          _approve(account_, msg.sender, decreasedAll
            _approve(account_, msg.sender, decreasedAll
413
                                                             413
    owance );
                                                                 owance_);
414
            _burn(account_, amount_);
                                                             414
                                                                         _burn(account_, amount_);
415
                                                             415
416 }
                                                             416 }
```