```
0x5ca
-----
assert(0 == msg.value)
$s4 = 0x4 + c[0x4]
t = c[$s4]
$s7 = $m
m = m + (0x20 + (0x20 * ((0x1f + $t) / 0x20)))
m[\$s7] = \$t
calldatacopy(0x20 + $s7, 0x20 + $s4, $t)
$s10 = $m
$s13 = 0x4 + c[0x44]
$s12 = c[$s13]
m = 0x20 + (m + (0x20 * ((0x1f + $s12) / 0x20)))
m[\$s10] = \$s12
$s3 = 0xff \& c[0x24]
calldatacopy(0x20 + $s10, 0x20 + $s13, $s12)
$s4 = $s10
$s5 = c[0x64]
$s6 = c[0x84]
if (! $s5){
 $s5 = block.number
$s9 = ad mask & (s[0xc] / 0x100)
m[0x20 + \$m] = 0x0
m[$m] = 0x5b7b72c1 << 0xe0
$s17 = 0x4 + $m
m[\$s17] = ad mask \& self
$s18 = 0x20 + $s17
m[\$s18] = \$s5
$s18 = 0x20 + $s18
$s19 = 0x20 + $s18
m[\$s19] = 0xff \& \$s3
$s19 = 0x20 + $s19
$s20 = 0x20 + $s19
m[\$s20] = \$s6
$s20 = 0x20 + $s20
m[\$s18] = \$s20 - \$s17
m[\$s20] = m[\$s7]
$s20 = 0x20 + $s20
t = m[$s7]
$s23 = $t
$s24 = $s20
$s25 = 0x20 + $s7
if (! 0x0 == $t){
 while (0x1) {
   m[\$s24] = m[\$s25]
   if ($s23 \le 0x20)
       break
   $s24 = 0x20 + $s24
   $s25 = 0x20 + $s25
 }
$s20 = $t + $s20
$s21 = 0x1f \& $t
if ($s21){
 $s22 = $s20 - $s21
 m[\$s22] = (! ((0x100 ** (0x20 - \$s21)) - 0x1)) \& m[\$s22]
 $s20 = 0x20 + $s22
m[\$s19] = \$s20 - \$s17
m[\$s20] = m[\$s4]
$s20 = 0x20 + $s20
t = m[\$s4]
$s23 = $t
$s24 = $s20
$s25 = 0x20 + $s4
if ($t){
 while (0x1) {
   m[\$s24] = m[\$s25]
   if (\$s23 \le 0x20)
       break
   $s24 = 0x20 + $s24
   $s25 = 0x20 + $s25
 }
$s20 = $t + $s20
$s21 = 0x1f \& $t
if ($s21){
 $s22 = $s20 - $s21
 m[\$s22] = (! ((0x100 ** (0x20 - \$s21)) - 0x1)) \& m[\$s22]
 $s20 = 0x20 + $s22
assert(extcodesize($s9))
assert(call(msg.gas - 0x2c6, $s9, 0x0, $m, $s20 - $m, $m, 0x20))
$s12 = m[$m]
m[0x4 + $m] = msq.sender
$s8 = $s12
$s12 = ad mask & $s12
assert(extcodesize($s12))
assert(call(msg.gas - 0x2c6, $s12, 0x0, $m, 0x24, $m, 0x0))
log2($m, 0x20, 0x86c875b377f900b07ce03575813022f05dd10ed7640b5282cf6d3c3fc352ade, ad mask & $s8)
m[$m] = ad mask & $s8
return($m, 0x20)
```