```
0x386
assert(0 == msg.value)
$s2 = $m
$s5 = c[0xa4]
$s6 = c[0x4 + $s5]
m = 0x20 + (m + (0x20 * ((0x1f + $s6) / 0x20)))
m[\$s2] = \$s6
$t = $s2
$s2 = c[0x4]
$s7 = $t
$s3 = ad mask \& c[0x24]
$s4 = ad mask \& c[0x44]
$t = $s5
$s5 = ad mask & c[0x64]
$s11 = $t
$t = $s6
$s6 = c[0x84]
calldatacopy(0x20 + $s7, 0x24 + $s11, $t)
$s15 = $m
$s18 = 0x4 + c[0xc4]
$s17 = c[$s18]
t = s18
$s18 = 0x20 + ($m + (0x20 * ((0x1f + $s17) / 0x20)))
m = 18
m[\$s15] = \$s17
calldatacopy(0x20 + $s15, 0x20 + $t, $s17)
$s11 = (ad mask \& s[0x0]) == msg.sender
if (! $s11){
 $s11 = (block.timestamp - s[0x1]) > 0x278d00
if ($s11){
 m[0x0] = $s2
 m[0x20] = 0x2
 assert(0 == (0xff \& s[sha3(0x0, 0x40)]))
 m[\$s18] = \$s2
 m[0x20 + $s18] = (ad mask & $s3) << 0x60
 m[0x34 + $s18] = (ad mask & $s4) << 0x60
 m[0x48 + $s18] = (ad_mask & $s5) << 0x60
 m[0x5c + $s18] = $s6
 $s11 = intcall1($s7, sha3($s18, 0x7c), $s4, 0x95d)
 assert($s11)
 m[0x4 + $s18] = ad mask & $s4
 m[0x24 + $s18] = ad mask & $s5
 m[0x44 + $s18] = $s6
 $s11 = ad mask \& $s3
 assert(extcodesize($s11))
 assert(call(msg.gas - 0x32, $s11, 0x0, $s18, 0x64, $s18, 0x0))
 m[0x0] = $s2
 m[0x20] = 0x2
 $s11 = sha3(0x0, 0x40)
 stop()
```