```
0xf2
--------
$s2 = c[0x4]
$s3 = c[0x24]
$s4 = c[0x44]
$s5 = c[0x64]
$s6 = c[0x84]
$s7 = c[0xa4]
$s8 = 0xff & c[0xc4]
m[0x0] = $s5
m[0x20] = 0x5
$s10 = sha3(0x0, 0x40)
$s9 = $s10
$s12 = 0x0
assert(0 == (ad_mask & (s[0x1 + $s10] >> 0x60)))
$s10 = msg.value
$s15 = $s3 > 0x1
if ($s15){
  $s15 = $s3 <= 0x64
}
assert($s15)
$s15 = $s10 >= 0x2386f26fc10000
if ($s15){
  $s15 = $s10 <= 0x3f870857a3e0e3800000
}
assert($s15)
$s15 = $s2 > 0x0
if ($s15){
  $s15 = $s2 < 0x10000000000
}
assert($s15)
$s15 = (0xff & $s8) >= 0x1b
if ($s15){
  $s15 = (0xff & $s8) <= 0x1c
}
assert($s15)
assert($s4 >= block.number)
$s18 = 0x20 + $m
m[$s18] = (0xffffffffff & $s4) << 0xd8
$s19 = 0x5 + $s18
m[$s19] = $s5
$s16 = 0x20 + $s19
m[$m] = ($s16 - $m) - 0x20
$t = $s16
$s16 = $m
$m = $t
$s17 = $t
$t = m[$s16]
$s21 = $t
$s22 = $s17
$s23 = 0x20 + $s16
while (0x1) {
  if ($s21 < 0x20)
        break
  m[$s22] = m[$s23]
  $s21 = 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe0 + $s21
  $s22 = 0x20 + $s22
  $s23 = 0x20 + $s23
}
$s24 = (0x100 ** (0x20 - $s21)) - 0x1
m[$s22] = (m[$s23] & (! $s24)) | (m[$s22] & $s24)
$s16 = sha3($m, ($t + $m) - $m)
m[$m] = 0x0
$m = 0x20 + $m
m[$m] = $s16
$s21 = 0x20 + $m
m[$s21] = 0xff & $s8
$s21 = 0x20 + $s21
m[$s21] = $s6
$s21 = 0x20 + $s21
m[$s21] = $s7
assert(0 == (0 == call(msg.gas, 0x1, 0x0, $m, ((0x20 + $s21) - $m), ($m - 0x20), 0x20)))
assert((ad_mask & m[0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe0 + $m]) == (ad_mask & s[0x3]))
if ($s3 <= 0x28){
  $s11 = (0x1041041041041041041041041041041041041041041041041041041041041041 & (0x2000000000100000000008000000000400000000020000000001 * $s2)) % 0x3f
  $s12 = $s2
} else {
  $s15 = $s2 > 0x0
  if ($s15){
    $s15 = $s2 <= $s3
  }
  assert($s15)
  $s11 = $s2
}
$s15,$s16 = intcall0($s11, $s3, $s10, 0x7ec)
$t = $s15
$s14 = $s16
assert($t <= ($s10 + s[0x2]))
$s16 = s[0x4]
$s18 = (0xffffffffffffffffffffffffffffffff & $s16) | ((0xffffffffffffffffffffffffffffffff & ($t + (0xffffffffffffffffffffffffffffffff & ($s16 >> 0x80)))) << 0x80)
$s15 = (0xffffffffffffffffffffffffffffffff00000000000000000000000000000000 & $s18) | (0xffffffffffffffffffffffffffffffff & ($s14 + (0xffffffffffffffffffffffffffffffff & $s18)))
s[0x4] = $s15
assert((0xffffffffffffffffffffffffffffffff & ((0xffffffffffffffffffffffffffffffff & ($s15 >> 0x80)) + (0xffffffffffffffffffffffffffffffff & $s15))) <= balance(self))
m[$m] = $s5
log1($m, 0x20, 0x5bdd2fc9902253015777690475b670d3872f32262eb1d47d9ba8000dad58f87)
s[$s9] = $s10
$t = $s9
$s9 = $s11
$s11 = 0x1 + $t
s[$s11] = (msg.sender << 0x60) | (0xffffffffffffffffffffffff & (((0xffffffffff & $s12) << 0x38) | (0xffffffffffffffffffffffffffffffffffffffff0000000000ffffffffffffff & (((0xffffffffff & block.number) << 0x10) | (0xffffffffffffffffffffffffffffffffffffffffffffffffff0000000000ffff & ((0x100 * (0xff & $s9)) | (0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff00ff & ((0xff & $s3) | (0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff00 & s[$s11])))))))))
stop()
```