```
0x436
--------
assert(0 == msg.value)
$s2 = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & c[0x4]
$s3 = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & c[0x24]
$s4 = c[0x44]
assert(msg.sender == (ad_mask & s[0xb]))
m[0x20 + $m] = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & $s2
m[0x34 + $m] = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & $s3
m[$m] = 0x28
$s9 = $m
$s11 = 0x48 + $m
$m = $s11
$s8 = $s9
$t = m[$s9]
$s9 = $s11
$s12 = $t
$s13 = $t
$s14 = $s9
$s15 = 0x20 + $s8
while (0x1) {
  if ($s13 < 0x20)
        break
  m[$s14] = m[$s15]
  $s13 = 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe0 + $s13
  $s14 = 0x20 + $s14
  $s15 = 0x20 + $s15
}
$s17 = (0x100 ** (0x20 - $s13)) - 0x1
m[$s14] = ($s17 & m[$s14]) | (m[$s15] & (! $s17))
m[0x0] = sha3($m, ($s9 + $s12) - $m)
m[0x20] = 0xa
$s12 = sha3(0x0, 0x40)
$s7 = 0xfffffffff & (s[0x1 + $s12] >> 0x10)
assert($s7 < block.number)
assert(block.number <= (0xfa + $s7))
assert($s4 == blockhash($s7))
$s13 = s[$s12]
$s14 = s[0x1 + $s12]
$t = $s14
$s14 = 0xff & $s14
$s15 = 0xff & ($t / 0x100)
$s16 = ad_mask & ($t >> 0x60)
assert($s13)
s[$s12] = 0x0
$s27 = 0x20 + $m
m[$s27] = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & $s2
$s28 = 0x14 + $s27
m[$s28] = $s4
$s28 = 0x20 + $s28
m[$s28] = 0xffffffffffffffffffffffffffffffffffffffff000000000000000000000000 & $s3
$s24 = 0x14 + $s28
m[$m] = ($s24 - $m) - 0x20
$t = $s24
$s24 = $m
$m = $t
$s25 = $t
$t = m[$s24]
$s28 = $t
$s29 = $t
$s30 = $s25
$s31 = 0x20 + $s24
while (0x1) {
  if ($s29 < 0x20)
        break
  m[$s30] = m[$s31]
  $s29 = 0xffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffe0 + $s29
  $s30 = 0x20 + $s30
  $s31 = 0x20 + $s31
}
$s32 = (0x100 ** (0x20 - $s29)) - 0x1
m[$s30] = (m[$s30] & $s32) | (m[$s31] & (! $s32))
$s27 = sha3($m, ($m + $s28) - $m)
assert($s14)
$s18 = $s27 % $s14
$s24,$s25 = intcall0($s15, $s14, $s13, 0x1d28)
$s25 = $s24
$s19 = $s24
$s21 = 0x0
$s22 = 0x0
if ($s14 <= 0x28){
  if (0xfffffffff & ((s[0x1 + $s12] >> 0x38) & (0x2 ** $s18))){
    $s21 = $s19
  }
  goto 0x1d74
} else {
  if ($s18 < $s15){
    $s21 = $s25
  }
}
$s25 = s[0x9]
s[0x9] = (0xffffffffffffffffffffffffffffffff & $s25) | ((0xffffffffffffffffffffffffffffffff & ((0xffffffffffffffffffffffffffffffff & ($s25 >> 0x80)) - $s19)) << 0x80)
if ($s13 >= s[0x2]){
  $s24 = s[0x3]
  assert($s14)
  assert($s24)
  if (! ($s27 / $s14) % $s24){
    $s25 = s[0x9]
    s[0x9] = 0xffffffffffffffffffffffffffffffff00000000000000000000000000000000 & $s25
    $s22 = 0xffffffffffffffffffffffffffffffff & $s25
  }
}
if ($s22 > 0x0){
  m[$m] = $s22
  log2($m, 0x20, 0xc388db0e8aa560a59633c094a0d0aa21322cd6234836fd5bac00fc5ae63b5783, ad_mask & $s16)
}
if ($s22 + $s21){
  $s26 = $s21 + $s22
} else {
  $s26 = 0x1
}

if (call(0x8fc * (0 == $s26), ad_mask & $s16, $s26, $m, 0x0, $m, 0x0)){
  m[$m] = $s21
  log2($m, 0x20, 0xd4f43975feb89f48dd30cabbb32011045be187d1e11c8ea9faa43efc35282519, ad_mask & $s16)
} else {
  m[$m] = $s26
  log2($m, 0x20, 0xac464fe4d3a86b9121261ac0a01dd981bfe0777c7c9d9c8f4473d31a9c0f9d2d, ad_mask & $s16)
}


stop()
```