```
0x175
--------
$s2 = c[0x4]
$s3 = c[0x24]
$s4 = c[0x44]
$s5 = ad_mask & c[0x64]
assert(0 == (0xff & (s[0x0] >> 0xa0)))
assert(block.timestamp <= 0x5aac5a7f)
assert($s2 <= 0x3)
assert($s3 <= 0x3)
assert($s4 <= 0x3)
$s6 = msg.sender
$s7 = 0x0
if (! (ad_mask & $s6) == (ad_mask & $s5)){
  $s7 = $s5
}
$s8 = msg.value
if ($s2 > 0x0){
  $s12 = intcall4($s7, $s2, 0x0, $s6, 0x4a3)
  assert($s8 >= $s12)
  $s8 = $s8 - $s12
}
if ($s3 > 0x0){
  $s12 = intcall4($s7, $s3, 0x2, $s6, 0x4ce)
  assert($s8 >= $s12)
  $s8 = $s8 - $s12
}
if ($s4 > 0x0){
  $s12 = intcall4($s7, $s4, 0x4, $s6, 0x4f9)
  assert($s8 >= $s12)
  $s8 = $s8 - $s12
}
assert(call(0x8fc * (0 == $s8), msg.sender, $s8, $m, 0x0, $m, 0x0))
if (ad_mask & $s7){
  m[0x0] = ad_mask & $s7
  m[0x20] = 0x5
  $s12 = s[sha3(0x0, 0x40)]
  $s20 = $s2 + $s12
  assert($s20 >= $s12)


  $s18 = $s3 + $s20
  assert($s18 >= $s20)


  $s16 = $s4 + $s18
  assert($s16 >= $s18)

  $s10 = $s16
  $s11 = $s16 / 0x5
  if ($s11 > 0x0){
    m[0x0] = ad_mask & $s7
    m[0x20] = 0x5
    s[sha3(0x0, 0x40)] = $s10 % 0x5
    m[0x20] = 0x6
    $s12 = intcall1($s11, s[sha3(0x0, 0x40)], 0x5cf)
    m[0x0] = ad_mask & $s7
    m[0x20] = 0x6
    s[sha3(0x0, 0x40)] = $s12
    $s12 = intcall1($s11, s[0x7], 0x5fb)
    s[0x7] = $s12
    m[$m] = $s11
    log2($m, (0x20 + $m) - $m, 0xd10165fd5b921c47fdb2555c3f7acdb72b54e3dc98d8bba85b66980434d58bb8, ad_mask & $s7)
    goto 0x65c
  } else {
    m[0x0] = ad_mask & $s7
    m[0x20] = 0x5
    s[sha3(0x0, 0x40)] = $s16
  }
}

stop()
```