```
0xb9d
--------
$s2 = ad_mask & c[0x4]
$s3 = 0xffffffffff & c[0x24]
$s4 = 0xffffffffff & c[0x44]
$s5 = 0xffffffffffffffffffffffffffffffff & c[0x64]
$s6 = c[0x84]
$s7 = 0xff & c[0xa4]
$s8 = c[0xc4]
$s9 = c[0xe4]
assert(0 == (0xff & (s[0xa] >> 0xa0)))
$s10 = 0 == (0xffffffffff & $s4)
if (! $s10){
  $s10 = intcall3($s4, msg.sender, 0x2bd2)
}
assert($s10)
m[0x0] = ad_mask & $s2
m[0x20] = 0xd
assert(ad_mask & s[sha3(0x0, 0x40)])
m[0x0] = 0xffffffffff & $s3
m[0x20] = 0xf
assert(0 == (ad_mask & s[sha3(0x0, 0x40)]))
assert((0xffffffffff & $s3) >= (0xffffffffff & s[0x10]))
assert((0xffffffffffffffffffffffffffffffff & $s5) <= msg.value)
$s10 = intcall6($s9, $s8, $s7, $s6, $s5, $s4, $s3, $s2, 0x2c6d)
assert($s10)
m[0x0] = 0xffffffffff & $s3
m[0x20] = 0xf
$s11 = sha3(0x0, 0x40)
$s13 = msg.sender
s[$s11] = (ad_mask & $s13) | (0xffffffffffffffffffffffff0000000000000000000000000000000000000000 & s[$s11])
m[$m] = 0xffffffffff & $s3
m[0x20 + $m] = ad_mask & $s13
log1($m, (0x40 + $m) - $m, 0x3b213e544d7183ac9b8a682aa7ad1d52d0b2c05c7aba9bf7bc025d7b3847b63e)
m[0x0] = ad_mask & $s2
m[0x20] = 0xd
$s11 = ad_mask & s[sha3(0x0, 0x40)]
m[$m] = 0x9652713e0000000000000000000000000000000000000000000000000000000000
m[0x4 + $m] = 0xffffffffff & $s4
m[0x24 + $m] = $s6
m[0x44 + $m] = ad_mask & msg.sender
assert(extcodesize($s11))
assert(call(msg.gas, $s11, $s5, $m, (0x64 + $m) - $m, $m, 0x0))
stop()
```