```
0x26f
--------
$s3 = ad_mask & c[0x4]
$s4 = $s3
$s2 = $s3
$s3 = c[0x24]
m[0x0] = ad_mask & $s2
m[0x20] = 0x2
$s6 = sha3(0x0, 0x40)
m[0x0] = $s3
m[0x20] = $s6
$s9 = msg.value
m[0x0] = ad_mask & $s2
m[0x20] = 0x2
$s17 = sha3(0x0, 0x40)
m[0x0] = $s3
m[0x20] = $s17
$s17 = sha3(0x0, 0x40)
$s11 = $s17
$s17 = intcall3($s17, 0x122e)
assert($s17)
$s17 = intcall4($s11, 0x1242)
$s12 = $s17
assert(0 == ($s9 < $s17))
$s13 = ad_mask & s[0x1 + $s11]
 = intcall0($s3, $s4, 0x1284)
if ($s12 > 0x0){
  assert(0x1)
  $s15 = $s12 - (($s12 * s[0x1]) / 0x2710)
  assert(call(0x8fc * (0 == $s15), ad_mask & $s13, $s15, $m, 0x0, $m, 0x0))
}
$s16 = $s9 - $s12
assert(call(0x8fc * (0 == $s16), msg.sender, $s16, $m, 0x0, $m, 0x0))
m[$m] = ad_mask & (ad_mask & s[0x1 + $s11])
$s24 = 0x20 + $m
m[$s24] = ad_mask & $s4
$s24 = 0x20 + $s24
m[$s24] = $s3
$s24 = 0x20 + $s24
m[$s24] = $s12
$s24 = 0x20 + $s24
m[$s24] = ad_mask & msg.sender
log1($m, (0x20 + $s24) - $m, 0xe231499744be2bdc17374b7cf3d7d092ded2ebed31d0cccc5931de65065594ef)
 = intcall2($s3, msg.sender, $s4, 0x895)
stop()
```