```
0x5c2
--------
$s2 = c[0x4]
assert(0x1 != (0xff & s[0x5]))
$s4 = msg.value
$s6 = s[0x9]
$s9 = s[0x8]
$s11 = intcall2(0x1, $s2, 0x155d)
$s12 = intcall2(0x1, $s2, 0x1568)
assert($s11)
assert(0x1)
$s6 = ((((($s4 * (0x64 - $s12)) / $s11) + $s4) * $s9) / 0x3e8) - $s4) > $s6
if (! $s6){
  $s6 = $s4 < s[0xb]
}
if (! $s6){
  $s6 = $s2 < 0x2
}
if (! $s6){
  $s6 = $s2 > 0x63
}
assert(0 == $s6)
$s7 = $m
m[$m] = 0x6
m[0x20 + $m] = 0x6e6573746564000000000000000000000000000000000000000000000000000000
$s8 = 0x40 + $m
$m = 0x1e0 + $s8
m[$s8] = 0x1b4
codecopy(0x20 + $s8, 0x26e8, 0x1b4)
$s9 = 0xffffffff & (s[0x5] / 0x100)
$s12 = 0 == (ad_mask & s[0x0])
if (! $s12){
  $s12 = intcall9(ad_mask & s[0x0], 0x1af6)
  $s12 = 0 == $s12
}
if ($s12){
  $s12 = intcall8(0x0, 0x1b07)

}
$s12 = ad_mask & s[0x0]
m[0x20 + $m] = 0x0
m[$m] = 0x38cc4831 << 0xe0
assert(extcodesize($s12))
assert(call(msg.gas - 0x2c6, $s12, 0x0, $m, (0x4 + $m) - $m, $m, 0x20))
if (! (ad_mask & m[$m]) == (ad_mask & s[0x1])){
  $s12 = ad_mask & s[0x0]
  m[0x20 + $m] = 0x0
  m[$m] = 0x38cc4831 << 0xe0
  assert(extcodesize($s12))
  assert(call(msg.gas - 0x2c6, $s12, 0x0, $m, (0x4 + $m) - $m, $m, 0x20))
  s[0x1] = (ad_mask & m[$m]) | (0xffffffffffffffffffffffff0000000000000000000000000000000000000000 & s[0x1])
}
$s12 = ad_mask & s[0x1]
m[0x20 + $m] = 0x0
m[$m] = 0x2ef3accc << 0xe0
$s16 = 0x4 + $m
$s18 = 0x20 + $s16
m[$s18] = $s9
$s18 = 0x20 + $s18
m[$s16] = $s18 - $s16
m[$s18] = m[$s7]
$s18 = 0x20 + $s18
$t = m[$s7]
$s19 = 0x20 + $s7
$s24 = 0x0
while (0x1) {
  if ($s24 >= $t)
      break
  m[$s24 + $s18] = m[$s19 + $s24]
  $s24 = 0x20 + $s24
}
$s19 = $t
$t = $s18
$s18 = $s19
$s19 = $s19 + $t
$t = $s18
$s18 = $s19
$s19 = 0x1f & $t
if ($s19){
  $s20 = $s18 - $s19
  m[$s20] = (! ((0x100 ** (0x20 - $s19)) - 0x1)) & m[$s20]
  $s18 = 0x20 + $s20
}
assert(extcodesize($s12))
assert(call(msg.gas - 0x2c6, $s12, 0x0, $m, $s18 - $m, $m, 0x20))
$s11 = m[$m]
if ($s11 > 0xde0b6b3a7640000 + ($s9 * gasprice())){
  $s10 = 0x0
} else {
  $s12 = ad_mask & s[0x1]
  m[0x20 + $m] = 0x0
  m[$m] = 0xc51be90f << 0xe0
  $s19 = 0x4 + $m
  m[$s19] = 0x0
  $s20 = 0x20 + $s19
  $s21 = 0x20 + $s20
  $s22 = 0x20 + $s21
  m[$s22] = $s9
  $s22 = 0x20 + $s22
  m[$s20] = $s22 - $s19
  m[$s22] = m[$s7]
  $s22 = 0x20 + $s22
  $t = m[$s7]
  $s23 = 0x20 + $s7
  $s28 = 0x0
  while (0x1) {
    if ($s28 >= $t)
        break
    m[$s28 + $s22] = m[$s23 + $s28]
    $s28 = 0x20 + $s28
  }
  $s23 = $t
  $t = $s22
  $s22 = $s23
  $s23 = $s23 + $t
  $t = $s22
  $s22 = $s23
  $s23 = 0x1f & $t
  if ($s23){
    $s24 = $s22 - $s23
    m[$s24] = (! ((0x100 ** (0x20 - $s23)) - 0x1)) & m[$s24]
    $s22 = 0x20 + $s24
  }
  m[$s21] = $s22 - $s19
  m[$s22] = m[$s8]
  $s22 = 0x20 + $s22
  $t = m[$s8]
  $s23 = 0x20 + $s8
  $s28 = 0x0
  while (0x1) {
    if ($s28 >= $t)
        break
    m[$s28 + $s22] = m[$s23 + $s28]
    $s28 = 0x20 + $s28
  }
  $s23 = $t
  $t = $s22
  $s22 = $s23
  $s23 = $s23 + $t
  $t = $s22
  $s22 = $s23
  $s23 = 0x1f & $t
  if ($s23){
    $s24 = $s22 - $s23
    m[$s24] = (! ((0x100 ** (0x20 - $s23)) - 0x1)) & m[$s24]
    $s22 = 0x20 + $s24
  }
  assert(extcodesize($s12))
  assert(call(msg.gas - 0x25ee, $s12, $s11, $m, $s22 - $m, $m, 0x20))
  $s10 = m[$m]
}
$s6 = $s10
m[0x0] = $s10
m[0x20] = 0x12
s[sha3(0x0, 0x40)] = $s6
m[0x20] = 0x16
s[sha3(0x0, 0x40)] = $s2
m[0x20] = 0x13
$s10 = msg.value
s[sha3(0x0, 0x40)] = $s10
m[0x20] = 0x10
$s7 = $s10
$s8 = sha3(0x0, 0x40)
s[$s8] = msg.sender | (0xffffffffffffffffffffffff0000000000000000000000000000000000000000 & s[$s8])
$s8 = s[0x8]
$s3 = $s6
$s10 = intcall2(0x1, $s2, 0x1679)
$s11 = intcall2(0x1, $s2, 0x1684)
assert($s10)
assert(0x3e8)
m[0x0] = $s6
m[0x20] = 0x18
$s7 = (((((msg.value * (0x64 - $s11)) / $s10) + $s7) * $s8) / 0x3e8) - $s7
s[sha3(0x0, 0x40)] = $s7
$s6 = intcall6($s7, s[0xd], 0x16c3)
s[0xd] = $s6
assert($s6 < s[0x7])
m[0x0] = $s3
m[0x20] = 0x13
$s9 = s[sha3(0x0, 0x40)]
m[0x20] = 0x18
$s6 = intcall6(s[sha3(0x0, 0x40)], $s9, 0x16fb)
m[0x0] = $s3
m[0x20] = 0x10
$s10 = s[sha3(0x0, 0x40)]
m[0x20] = 0x12
$s11 = s[sha3(0x0, 0x40)]
m[0x20] = 0x18
$s12 = s[sha3(0x0, 0x40)]
m[0x20] = 0x13
$s8 = s[sha3(0x0, 0x40)]
m[0x20] = 0x16
$s13 = s[sha3(0x0, 0x40)]
m[$m] = $s12
$s14 = 0x20 + $m
m[$s14] = $s8
$s14 = 0x20 + $s14
m[$s14] = $s13
log4($m, (0x20 + $s14) - $m, 0x1cb5bfc4e69cbacf65c8e05bdb84d7a327bd6bb4c034ff82359aefd7443775c4, $s11, ad_mask & $s10, $s6)
stop()
```