```
0x29d
--------
assert(0 == msg.value)
$s2 = ad_mask & c[0x4]
$s3 = c[0x24]
$s4 = 0xff & c[0x44]
$s5 = c[0x64]
$s6 = c[0x84]
m[0x0] = msg.sender
m[0x20] = 0x7
assert(0xff & s[sha3(0x0, 0x40)])
if (s[0x6] > 0x0){
  s[0x6] = 0x278d00 + block.timestamp
}
assert(0x1)
$s7 = (msg.gas / 0x3e8) * s[0x4]
m[0x0] = ad_mask & $s2
m[0x20] = 0x9
$s13 = s[sha3(0x0, 0x40)]
m[$m] = (ad_mask & $s2) << 0x60
m[0x14 + $m] = $s3
m[0x34 + $m] = $s13
$s11 = sha3($m, (0x54 + $m) - $m)
m[$m] = 0x0
$m = 0x20 + $m
m[0x20 + $m] = 0x0
m[$m] = $s11
m[0x20 + $m] = 0xff & $s4
m[0x40 + $m] = $s5
m[0x60 + $m] = $s6
assert(call(msg.gas - 0x646e, 0x1, 0x0, $m, (0x80 + $m) - $m, $m - 0x20, 0x20))
$s10 = m[$m - 0x20]
m[0x0] = ad_mask & $s2
m[0x20] = 0x9
$s11 = sha3(0x0, 0x40)
s[$s11] = 0x1 + s[$s11]
$s8 = $s10
$s15 = 0x2710 * $s3
$s16 = 0 == $s3
if (! $s16){
  assert($s3)
  $s16 = 0x2710 == ($s15 / $s3)
}
assert($s16)


$s14 = $s7 + $s15
$s15 = $s14 < $s15
$t = $s15
$s15 = 0 == $s15
if (! $t){
  assert($s14 >= $s7)
} else {
  assert($s15)
}

m[0x0] = ad_mask & $s10
m[0x20] = 0x2
$s10 = intcall3($s14, s[sha3(0x0, 0x40)], 0xb67)
m[0x0] = ad_mask & $s8
m[0x20] = 0x2
s[sha3(0x0, 0x40)] = $s10
$s10 = intcall3($s14, s[0x1], 0xb8d)
s[0x1] = $s10
$s10 = ad_mask & s[0x5]
m[0x20 + $m] = 0x0
m[$m] = 0xa9059cbb0000000000000000000000000000000000000000000000000000000000000000
m[0x4 + $m] = ad_mask & $s2
m[0x24 + $m] = $s3
assert(extcodesize($s10))
assert(call(msg.gas - 0x2c6, $s10, 0x0, $m, (0x44 + $m) - $m, $m, 0x20))
assert(m[$m])
m[$m] = ad_mask & $s8
m[0x20 + $m] = ad_mask & $s2
m[0x40 + $m] = $s3
log1($m, (0x60 + $m) - $m, 0x2717ead6b9200dd235aad468c9809ea400fe33ac69b5bfaa6d3e90fc922b6398)
stop()
```