```
0xd87
--------
$s2 = 0xffffffffff & c[0x4]
$s3 = 0xffffffffff & c[0x24]
assert(0 == (0xff & (s[0xa] >> 0xa0)))
$s7 = intcall3($s2, msg.sender, 0x34a0)
assert($s7)
m[0x0] = 0xffffffffff & $s2
m[0x20] = 0x1
$s10 = s[sha3(0x0, 0x40)]
m[0x0] = 0xffffffffff & $s3
$s13 = ad_mask & $s10
$s11 = $s13
$s13 = $s13 == (ad_mask & s[sha3(0x0, 0x40)])
if (! $s13){
  m[0x0] = 0xffffffffff & $s3
  m[0x20] = 0x4
  $s7 = (ad_mask & $s11) == (ad_mask & s[sha3(0x0, 0x40)])
} else {
  $s7 = $s13
}
assert($s7)
$s7 = msg.value
$s8 = intcall2($s3, $s2, 0x34cb)
assert($s8 <= $s7)
$s8 = 0xffffffffff & $s2
assert($s8 < s[0x0])
m[0x0] = 0x0
$s5 = (0x2 * $s8) + sha3(0x0, 0x20)
$s9 = $m
$m = 0x100 + $m
m[$s9] = s[$s5]
$s10 = s[0x1 + $s5]
m[0x20 + $s9] = 0xffffffffff & $s10
m[0x40 + $s9] = 0xffffffffff & ($s10 >> 0x28)
m[0x60 + $s9] = 0xffffffffff & ($s10 >> 0x50)
$t = $s10
$s8 = $t
m[0x80 + $s9] = 0xffffffffff & ($s10 >> 0x78)
m[0xa0 + $s9] = 0xffff & ($t >> 0xa0)
m[0xc0 + $s9] = 0xffff & ($s8 >> 0xb0)
m[0xe0 + $s9] = 0xffffffffffffffff & ($s8 >> 0xc0)
$s7 = intcall15($s9, 0x358f)
assert($s7)
$s8 = 0xffffffffff & $s3
assert($s8 < s[0x0])
m[0x0] = 0x0
$s6 = (0x2 * $s8) + sha3(0x0, 0x20)
$s9 = $m
$m = 0x100 + $m
m[$s9] = s[$s6]
$s10 = s[0x1 + $s6]
m[0x20 + $s9] = 0xffffffffff & $s10
m[0x40 + $s9] = 0xffffffffff & ($s10 >> 0x28)
m[0x60 + $s9] = 0xffffffffff & ($s10 >> 0x50)
$t = $s10
$s8 = $t
m[0x80 + $s9] = 0xffffffffff & ($s10 >> 0x78)
m[0xa0 + $s9] = 0xffff & ($t >> 0xa0)
m[0xc0 + $s9] = 0xffff & ($s8 >> 0xb0)
m[0xe0 + $s9] = 0xffffffffffffffff & ($s8 >> 0xc0)
$s7 = intcall15($s9, 0x3653)
assert($s7)
$s7 = intcall22($s3, $s6, $s2, $s5, 0x366a)
assert($s7)
$s7 = intcall17($s3, $s2, 0x367f)

m[$m] = 0xffffffffff & $s7
return($m, (0x20 + $m) - $m)
```