```
0x664
--------
assert(0 == msg.value)
assert(msg.sender == (ad_mask & s[0x2]))
$s3 = 0x0
$s4 = 0x0
while (0x1) {
  if ($s4 >= s[0x11])
        break
  m[0x0] = $s4
  m[0x20] = 0x10
  if (s[0x1 + sha3(0x0, 0x40)] < block.timestamp){
    m[0x0] = $s4
    m[0x20] = 0x10
    $s3 = $s3 + s[sha3(0x0, 0x40)]
    m[0x0] = $s4
    m[0x20] = 0x10
    $s8 = sha3(0x0, 0x40)
    $s7 = $s8
    m[0x0] = ad_mask & s[0x0]
    m[0x20] = 0x4
    $s8 = intcall1(s[$s8], s[sha3(0x0, 0x40)], 0x20bf)
    m[0x0] = ad_mask & s[0x0]
    m[0x20] = 0x4
    s[sha3(0x0, 0x40)] = $s8
    m[$m] = ad_mask & (ad_mask & s[0x0])
    $s13 = 0x20 + $m
    m[$s13] = s[$s7]
    $s13 = 0x20 + $s13
    m[$s13] = s[0x1 + $s7]
    log1($m, (0x20 + $s13) - $m, 0xc34f1fead36337f1ed421262dd3660824fdc849a066c9ae4699ffcd1b6e0ba50)
    m[0x0] = s[0x11] - 0x1
    m[0x20] = 0x10
    $s8 = sha3(0x0, 0x40)
    m[0x0] = $s4
    m[0x20] = 0x10
    $s9 = sha3(0x0, 0x40)
    s[$s9] = s[$s8]
    s[0x1 + $s9] = s[0x1 + $s8]
    m[0x0] = s[0x11] - 0x1
    m[0x20] = 0x10
    $s8 = sha3(0x0, 0x40)
    s[$s8] = 0x0
    s[0x1 + $s8] = 0x0
    s[0x11] = s[0x11] - 0x1
    goto 0x162f
  } else {
    $s4 = 0x1 + $s4
  }

}
m[$m] = $s3
return($m, (0x20 + $m) - $m)
```