```
0x3e0
--------
assert(0 == msg.value)
$s2 = 0x0
$s3 = 0x0
while (0x1) {
  if ($s3 >= s[0x8])
        break
  m[0x0] = $s3
  m[0x20] = 0x7
  if (block.timestamp >= s[0x2 + sha3(0x0, 0x40)]){
    m[0x0] = $s3
    m[0x20] = 0x7
    $s2 = $s2 + s[0x1 + sha3(0x0, 0x40)]
    m[0x0] = $s3
    m[0x20] = 0x7
    $s7 = sha3(0x0, 0x40)
    $s6 = $s7
    m[0x0] = ad_mask & s[$s7]
    m[0x20] = 0x4
    $s7 = intcall1(s[0x1 + $s7], s[sha3(0x0, 0x40)], 0x1e35)
    m[0x0] = ad_mask & s[$s6]
    m[0x20] = 0x4
    s[sha3(0x0, 0x40)] = $s7
    m[$m] = s[0x1 + $s6]
    $s12 = 0x20 + $m
    m[$s12] = s[0x2 + $s6]
    log2($m, (0x20 + $s12) - $m, 0x1f6c0a9bf76af8bc82d309c3041a10d100dafe019e5271d0e418fc57ffab3ab0, ad_mask & s[$s6])
    m[0x0] = s[0x8] - 0x1
    m[0x20] = 0x7
    $s7 = sha3(0x0, 0x40)
    m[0x0] = $s3
    m[0x20] = 0x7
    $s8 = sha3(0x0, 0x40)
    s[$s8] = (ad_mask & (ad_mask & s[$s7])) | (0xffffffffffffffffffffffff0000000000000000000000000000000000000000 & s[$s8])
    s[0x1 + $s8] = s[0x1 + $s7]
    s[0x2 + $s8] = s[0x2 + $s7]
    m[0x0] = s[0x8] - 0x1
    m[0x20] = 0x7
    $s7 = sha3(0x0, 0x40)
    s[$s7] = 0xffffffffffffffffffffffff0000000000000000000000000000000000000000 & s[$s7]
    s[0x1 + $s7] = 0x0
    s[0x2 + $s7] = 0x0
    s[0x8] = s[0x8] - 0x1
    goto 0x1281
  } else {
    $s3 = 0x1 + $s3
  }

}
m[$m] = $s2
return($m, (0x20 + $m) - $m)
```