```
0x7b5
--------
assert(0 == msg.value)
$s2 = 0xff & c[0x4]
$t = 0xffffffffffffffff & c[0x24]
$s4 = c[0x44]
m[0x0] = 0xffffffffffffffff & $t
m[0x20] = 0x4
$s6 = 0x3 + sha3(0x0, 0x40)
assert(0x1)
assert($s2 <= 0x4)
if (0x0 == $s2){
  m[0x0] = 0xffffffff & $t
  m[0x20] = 0x3
  $s6 = 0x1 + sha3(0x0, 0x40)
} else {
  assert(0x1)
  assert($s2 <= 0x4)
  if (0x1 == $s2){
    m[0x0] = 0xffffffff & $t
    m[0x20] = 0x3
    $s6 = 0x2 + sha3(0x0, 0x40)
  } else {
    assert(0x1)
    assert($s2 <= 0x4)
    if (0x2 == $s2){
      m[0x0] = 0xffffffff & $t
      m[0x20] = 0x3
      $s6 = 0x3 + sha3(0x0, 0x40)
    } else {
      assert(0x1)
      assert($s2 <= 0x4)
      if (0x4 == $s2){
        m[0x0] = 0xffffffffffffffff & $t
        m[0x20] = 0x4
        $s6 = 0x4 + sha3(0x0, 0x40)
      }

    }

  }

}
if ($s4 >= s[$s6]){
  $s5 = 0x0
} else {
  assert($s4 < s[$s6])
  m[0x0] = $s6
  $s5 = 0xff & (s[($s4 / 0x20) + sha3(0x0, 0x20)] / (0x100 ** ($s4 % 0x20)))
}
m[$m] = 0xff & $s5
return($m, (0x20 + $m) - $m)
```