```
0x23b
--------
assert(0 == msg.value)
$s2 = c[0x4]
$s4 = intcall1(0xb76)
m[0x0] = $s2
m[0x20] = 0xc
$s5 = sha3(0x0, 0x40)
$s6 = $m
$m = 0x120 + $m
m[$s6] = ad_mask & s[$s5]
m[0x20 + $s6] = s[0x1 + $s5]
m[0x40 + $s6] = s[0x2 + $s5]
m[0x60 + $s6] = s[0x3 + $s5]
$s8 = s[0x4 + $s5]
m[0x80 + $s6] = 0xffffffffffffffff & $s8
m[0xa0 + $s6] = 0xff & ($s8 >> 0x40)
m[0xc0 + $s6] = ad_mask & ($s8 >> 0x48)
m[0xe0 + $s6] = ad_mask & s[0x5 + $s5]
$t = $s5
$s5 = $s6
m[0x100 + $s5] = s[0x6 + $t]
$s8 = 0x0
if (block.timestamp > 0xffffffffffffffff & m[0x80 + $s5]){
  $s8 = block.timestamp - (0xffffffffffffffff & m[0x80 + $s5])
}
$s10 = m[0x20 + $s5]
$s11 = m[0x40 + $s5]
$s12 = m[0x60 + $s5]
if ($s8 >= $s12){
  $s14 = $s11
} else {
  assert($s12)
  $s14 = $s10 + ((($s11 - $s10) * $s8) / $s12)
}




m[$m] = $s14
return($m, (0x20 + $m) - $m)
```