```
0x76d
_ _ _ _ _ _ _ _
if (msg.value) goto 0x0
$s2 = $m
$s5 = c[0x64]
$s6 = c[0x4 + $s5]
m = 0x20 + (m + (0x20 * ((0x1f + $s6) / 0x20)))
m[\$s2] = \$s6
$t = $s2
$s2 = ad mask \& c[0x4]
$s7 = $t
s3 = c[0x24]
$s4 = c[0x44]
calldatacopy(0x20 + \$t, 0x24 + \$s5, \$s6)
$s6 = ad mask \& c[0x84]
$s9 = msq.sender
$s10 = intcall5($s6, 0x1678)
if (ad mask & $s10 == $s9){
 m[0x\overline{20} + \$m] = 0x0
 m[0x4 + $m] = ad mask & $s2
 m[0x24 + $m] = $s3
 m[0x44 + $m] = $s4
  m[0x84 + $m] = ad mask & $s6
 m[0x64 + $m] = 0xa0
  m[0xa4 + $m] = m[$s7]
  $s9 = ad mask \& s[0x0]
  $s21 = m[$s7]
  $s18 = 0xc4 + $m
  $s20 = $s21
  $s22 = $s18
  $s23 = 0x20 + $s7
  if ($s21){
   while (0x1) {
     m[\$s22] = m[\$s23]
     if (\$s21 \le 0x20)
       break
     $s22 = 0x20 + $s22
     $s23 = 0x20 + $s23
   }
  $s18 = $s20 + $s18
  $s19 = 0x1f \& $s20
  if ($s19){
   $s20 = $s18 - $s19
   m[\$s20] = (! ((0x100 ** (0x20 - \$s19)) - 0x1)) \& m[\$s20]
   $s18 = 0x20 + $s20
  if (0 == extcodesize($s9)) goto 0x0
  if (0 == call((msg.gas - 0x32), $s9, 0x0, $m, ($s18 - $m), $m, 0x20)) goto 0x0
  s1 = m[sm]
  goto 0x31d
} else {
  $s1 = 0x0
m[$m] = $s1
return($m, 0x20)
```