```
0x601
--------
assert(0 == msg.value)
$s2 = c[0x4]
$s3 = c[0x24]
$s4 = 0x0
$s5 = 0x0
$s6 = 0x0
$s7 = 0x0
$s8 = 0x0
```

```
0x20a3
--------
$s13 = $s8 < $s3
if ($s13){
  $s13 = ($s2 + $s8) < s[0x3]
}
if (0 == $s13) goto 0x21a9
```

```
0x20bb
--------
$s15 = $s8 + $s2
assert($s15 < s[0x3])
m[0x0] = 0x3
$s13,$s14,$s15,$s16 = intcall4(block.timestamp, ad_mask & s[sha3(0x0, 0x20) + $s15], 0x20f6)
$s11 = $s15
$s10 = $s14
$s15 = $s8 + $s2
assert($s15 < s[0x3])
m[0x0] = 0x3
$s14 = ad_mask & s[sha3(0x0, 0x20) + $s15]
$s15 = intcall2(block.timestamp, s[0xe], 0x2149)

$s16 = 0x0
$s17 = 0x0
if (! $s15){
  $s15 = block.timestamp
}
if (s[0x8]){
  $s15 = s[0x8]
}
$s20 = 0x0
$s21 = 0x0
while (0x1) {
  m[0x0] = ad_mask & $s14
  m[0x20] = 0x2
  if ($s21 >= s[sha3(0x0, 0x40)])
        break
  $s26 = s[0x6]
  m[0x0] = ad_mask & $s14
  m[0x20] = 0x2
  $s27 = sha3(0x0, 0x40)
  assert($s21 < s[$s27])
  m[0x0] = $s27
  $s27 = s[(0x2 * $s21) + sha3(0x0, 0x20)]
  assert($s26)
  m[0x0] = ad_mask & $s14
  m[0x20] = 0x2
  $s29 = sha3(0x0, 0x40)
  $t = $s26
  $s26 = s[$s29]
  $s30 = $t
  $t = $s29
  $s29 = $s30
  $s30 = $t
  $s22 = $s27 / $s29
  $t = $s26
  $s26 = s[0x7]
  assert($s21 < $t)
  m[0x0] = $s30
  $s27 = intcall3(s[0x1 + ((0x2 * $s21) + sha3(0x0, 0x20))], $s15, 0xc1f)
  assert($s26)
  $s26 = intcall1($s27 / $s26, $s22, 0xc3b)
  m[0x0] = ad_mask & $s14
  m[0x20] = 0x2
  $s28 = sha3(0x0, 0x40)
  $t = $s26
  assert($s21 < s[$s28])
  m[0x0] = $s28
  $s26 = intcall1(s[0x5], s[(0x2 * $s21) + sha3(0x0, 0x20)], 0xc93)
  $s25 = $s26
  if ($t >= $s26){
    $s26 = intcall2($s25, $s16, 0xcac)
    $s16 = $s26
    goto 0xcc6
  } else {
    $s26 = intcall2($t, $s16, 0xcc3)
    $s16 = $s26
  }
  $s26 = intcall2($s25, $s20, 0xcd6)
  m[0x0] = ad_mask & $s14
  m[0x20] = 0x2
  $s27 = sha3(0x0, 0x40)
  $s20 = $s26
  assert($s21 < s[$s27])
  m[0x0] = $s27
  $s26 = intcall2(s[(0x2 * $s21) + sha3(0x0, 0x20)], $s17, 0xd2c)
  $s17 = $s26
  $s21 = 0x1 + $s21
}
m[0x0] = ad_mask & $s14
m[0x20] = 0x2
$s26 = intcall2(s[0x2 + sha3(0x0, 0x40)], $s16, 0xd74)
$s16 = $s26
if ($s26 > $s20){
  $s16 = $s20
}
m[0x0] = ad_mask & $s14
m[0x20] = 0x2
$s26 = intcall2(s[0x1 + sha3(0x0, 0x40)], $s16, 0xdbc)
$t = $s14
$s14 = $s26
m[0x0] = ad_mask & $t
m[0x20] = 0x2
$s13 = intcall2($s13, $s4, 0x2166)
$s4 = $s13
$s13 = intcall2($s14, $s5, 0x2178)
$s5 = $s13
$s13 = intcall2($s10, $s6, 0x218a)
$s6 = $s13
$s13 = intcall2($s11, $s7, 0x219c)
$s7 = $s13
$s8 = 0x1 + $s8
goto 0x20a3
```

```
0x21a9
--------

m[$m] = $s4
m[0x20 + $m] = $s5
m[0x40 + $m] = $s6
m[0x60 + $m] = $s7
return($m, 0x80)
```