```
0x14c
--------
assert(0 == msg.value)
$s2 = c[0x4]
$s3 = c[0x24]
$s4 = c[0x44]
$s7 = c[0x64]
$s6 = c[0x84]
assert(msg.sender == (ad_mask & s[0x0]))
m[$m] = ((block.timestamp * block.number) + $s2) + $s4
$s9 = 0x0
$s10 = 0x0
assert(0x1)
$s8 = sha3($m, ((0x20 + $m) - $m)) / 0x64
if ($s3 <= $s7){
  $s15 = 0x1 + $s7
} else {
  $s15 = 0x1 + $s3
}
$s12 = $s15
$s11 = $s15
if ($s15 <= 0x190){
  $s15 = $s12
} else {
  $s15 = 0x190
}
assert(0x1)
$s12 = ((s[0x42] + (0x5 * $s15)) + ($s8 % 0x15)) << 0x64
$s13 = 0x2
assert(0x1)
$s14 = $s8 % 0x64
assert(0x1)
$s8 = $s8 / 0x64
assert(0x1)
$s15 = ($s8 % 0x64) < 0x2
if ($s15){
  assert(0x1)
  $s15 = 0x0 == ($s11 % 0x5)
}
if ($s15){
  $s13 = 0x1
}
if ($s14 < 0x2){
  $s11 = 0x0
  while (0x1) {
    if ($s11 >= 0xe)
        break
    if ($s11 > 0x0){
      $s9 = 0x10 * $s9
      $s10 = 0x10 * $s10
    }
    assert(0x1)
    $s14 = $s8 % 0x64
    assert(0x1)
    $s8 = $s8 / 0x64
    assert(0x1)
    if (0x0 == $s14 % 0x4){
      $s9 = $s9 + (0xf & ($s2 / (0x2 ** (0x4 * (0xd - $s11)))))
    } else {
      assert(0x1)
      if (0x1 == $s14 % 0x4){
        $s9 = $s9 + (0xf & ($s4 / (0x2 ** (0x4 * (0xd - $s11)))))
      } else {
        assert(0x1)
        if (0x2 == $s14 % 0x4){
          $s9 = $s9 + (0xf & ($s2 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
        } else {
          $s9 = $s9 + (0xf & ($s4 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
        }

      }

    }
    assert(0x1)
    $s14 = $s8 % 0x64
    assert(0x1)
    $s8 = $s8 / 0x64
    assert(0x1)
    if (0x0 == $s14 % 0x4){
      $s10 = $s10 + (0xf & ($s2 / (0x2 ** (0x4 * (0xd - $s11)))))
    } else {
      assert(0x1)
      if (0x1 == $s14 % 0x4){
        $s10 = $s10 + (0xf & ($s4 / (0x2 ** (0x4 * (0xd - $s11)))))
      } else {
        assert(0x1)
        if (0x2 == $s14 % 0x4){
          $s10 = $s10 + (0xf & ($s2 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
        } else {
          $s10 = $s10 + (0xf & ($s4 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
        }

      }

    }
    $s11 = 0x1 + $s11
  }
  goto 0x7b0
} else {
  $s11 = 0x0
  while (0x1) {
    if ($s11 >= 0xe)
        break
    if ($s11 > 0x0){
      $s9 = 0x10 * $s9
      $s10 = 0x10 * $s10
    }
    assert(0x1)
    $s14 = $s8 % 0xa
    assert(0x1)
    $s8 = $s8 / 0xa
    assert(0x1)
    if (0x0 == $s14 % 0x2){
      $s9 = $s9 + (0xf & ($s2 / (0x2 ** (0x4 * (0xd - $s11)))))
    } else {
      $s9 = $s9 + (0xf & ($s4 / (0x2 ** (0x4 * (0xd - $s11)))))
    }
    assert(0x1)
    $s14 = $s8 % 0xa
    assert(0x1)
    $s8 = $s8 / 0xa
    assert(0x1)
    if (0x0 == $s14 % 0x2){
      $s10 = $s10 + (0xf & ($s2 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
    } else {
      $s10 = $s10 + (0xf & ($s4 / (0x2 ** (0x80 + (0x4 * (0xd - $s11))))))
    }
    $s11 = 0x1 + $s11
  }

}
$s9 = $s9 + (((($s10 << 0x80) + $s12) + ($s13 << 0x7c))
$s12 = $s3 + $s7
$s19 = $s12
$s22 = 0x0
$s23 = 0x0
$s24 = 0x0
$s25 = 0x0
$s26 = 0x0
while (0x1) {
  if ($s26 >= 0x5)
      break
  if (! 0x0 == (0xff & ($s2 / (0x2 ** (0x8 * (0x1f - $s26))))){
    $s22 = 0x1 + $s22
    $s24 = (0x100 * $s24) + (0xff & ($s2 / (0x2 ** (0x8 * (0x1f - $s26))))
  }
  $s26 = 0x1 + $s26
}
$s26 = 0x0
while (0x1) {
  if ($s26 >= 0x5)
      break
  if (! 0x0 == (0xff & ($s4 / (0x2 ** (0x8 * (0x1f - $s26))))){
    $s22 = 0x1 + $s22
    $s24 = (0x100 * $s24) + (0xff & ($s4 / (0x2 ** (0x8 * (0x1f - $s26))))
  }
  $s26 = 0x1 + $s26
}
assert(0x1)
assert(0x1)
$s16 = $s8 / 0xa
if (($s8 % 0xa) >= $s22) goto 0x10e0
```

```
0xf38
--------
$s28 = 0x1
if ($s22 > 0x0){
  if (0x0 > 0x3){
    assert(0x1)
    $s28 = 0x0
  }
  goto 0xf7b
} else {
  if ($s22 > 0x3){
    assert(0x1)
    $s28 = $s22 / 0x2
  }

}
```

```
0xf7c
--------
if ($s28 >= 0x5) goto 0xfd5
```

```
0xf86
--------
$s30 = 0xa ** (0x1 + $s28)
assert($s30)
$s27 = $s16 % $s30
$s30 = 0xa ** (0x1 + $s28)
assert($s30)
$s16 = $s16 / $s30
if ($s27 >= ((0x1 + $s12) * ($s22 * $s22))) goto 0xfcb
```

```
0xfc1
--------
$s28 = 0x1 + $s28
goto 0xf7c
```

```
0xfcb
--------
goto 0xfd5
```

```
0xfd5
--------
$s19 = 0x0
$s26 = 0x0
```

```
0xfde
--------
$s30 = $s26 < $s28
if ($s30){
  $s30 = $s24 > 0x0
}
if (0 == $s30) goto 0x10df
```

```
0xff3
--------
assert(0x1)
$s27 = $s16 % 0x64
assert(0x1)
$s16 = $s16 / 0x64
$s32 = $s22 + $s23
assert($s32)
$s29 = 0xff & ($s24 / (0x2 ** (0x8 * ($s27 % $s32))))
$s25 = $s25 + ((0x2 ** (0x20 - (0x8 * $s26))) * $s29)
$s19 = $s19 + (0x2 ** (0x10 - (0x4 * $s26)))
$s27 = 0x0
while (0x1) {
  if ($s27 == ($s22 + $s23))
      break
  if (0xff & ($s24 / (0x2 ** (0x8 * $s27))) == $s29){
    $s24 = ((0x2 ** (0x8 * $s27)) * ($s24 / (0x2 ** (0x8 * (0x1 + $s27))))) + (((0x2 ** (0x8 * $s27)) - 0x1) & $s24)
    if ($s22 > 0x0){
      $s22 = $s22 - 0x1
    } else {
      $s23 = $s23 - 0x1
    }
    goto 0x10cd
  } else {
    $s27 = 0x1 + $s27
  }

}
$s26 = 0x1 + $s26
goto 0xfde
```

```
0x10df
--------
```

```
0x10e0
--------
$s9 = $s9 + (($s25 << 0xd8) + ($s19 << 0x50))
if (0x1 == $s13){
  $s9 = $s9 + ($s6 << 0x74)
}
$s15 = 0x1 == ($s2 >> 0x7c)
if (! $s15){
  $s15 = 0x1 == ($s4 >> 0x7c)
}
if ($s15){
  $s13 = 0x1
}
if (0x1 == $s13){
  $s28,$s29 = intcall0(0x0, 0x1117)
  $s20 = $s29
  $s19 = $s28
  goto 0x125e
} else {
  m[$m] = ((block.timestamp * block.number) + $s2) + $s4
  $s21 = sha3($m, ((0x20 + $m) - $m)
  $s26 = ((0x3ff & ($s2 >> 0xc6)) << 0xa) + (0x3ff & ($s4 >> 0xc6))
  $s27 = ((0x3ff & ($s2 >> 0xbc)) << 0xa) + (0x3ff & ($s4 >> 0xbc))
  $s24 = ((0x3ff & ($s2 >> 0x46)) << 0xa) + (0x3ff & ($s4 >> 0x46))
  $s25 = ((0x3ff & ($s2 >> 0x3c)) << 0xa) + (0x3ff & ($s4 >> 0x3c))
  assert(0x1)
  if ($s21 % 0x64 < 0x1){
    $s24 = $s24 + ($s26 << 0x14)
    $s25 = $s25 + ($s27 << 0x14)
  }
  $s28 = intcall2($s25, $s24, $s21, 0x1237)
  $s22 = $s28
  assert(0x1)
  $s28 = intcall2($s27, $s26, $s21 / 0xf4240, 0x1255)
  $s20 = $s28
  $s19 = $s22
}
m[$m] = $s9 + (($s19 << 0x3c) + ($s20 << 0xbc))
return($m, (0x20 + $m) - $m)
```