args $s10 $s9 $s8

rets $s8

```
0x1e83
--------
$s15 = 0x0
$s16 = intcall7($s9, 0x1e96)
$s12 = $s16
if (! $s16){
  $s16 = intcall3($s9, 0x1ea8)
  goto 0x1fc2
} else {
  $s13 = 0x0
  $s14 = 0x0
  while (0x1) {
    if ($s14 >= $s12)
        break
    m[0x0] = ad_mask & $s9
    m[0x20] = 0xf
    $s16 = sha3(0x0, 0x40)
    assert($s14 < s[$s16])
    m[0x0] = $s16
    $s19 = (0x3 * $s14) + sha3(0x0, 0x20)
    $s21 = $m
    $m = 0xe0 + $m
    m[$s21] = ad_mask & s[$s19]
    m[0x20 + $s21] = s[0x1 + $s19]
    $t = $s19
    $s19 = $s21
    $s21 = s[0x2 + $t]
    m[0x40 + $s19] = 0xffffffffffffffff & $s21
    m[0x60 + $s19] = 0xffffffffffffffff & ($s21 >> 0x40)
    $t = $s21
    m[0x80 + $s19] = 0xffffffffffffffff & ($s21 >> 0x80)
    m[0xa0 + $s19] = 0xff & ($t >> 0xc0)
    m[0xc0 + $s19] = 0xff & ($t >> 0xc8)
    $s18 = intcall5($s10, $s19, 0x1f85)

    $s20 = $s18 + $s13
     = intcall0($s20 >= $s13, 0x2838)

    $s13 = $s20
    $t = $s14
    $s14 = $s15
    $s15 = 0x1 + $t
    $t = $s14
    $s14 = $s15
    $s15 = $t
  }
  $s17 = intcall3($s9, 0x1fa4)

   = intcall0($s13 <= $s17, 0x298c)
  $s16 = $s17 - $s13

  $s25 = intcall14(block.number, $s9, 0x17f1)



  if ($s16 >= $s25){
    $s20 = $s25
  } else {
    $s20 = $s16
  }
  $s16 = $s20

}
$s8 = $s16
intret()
```