

```
0x623
-----
assert(0 == msg.value)
$s2 = intcall4(0xf51)
$s4 = intcall7(0xf5b)
assert($s4 < s[0x2])
m[0x0] = 0x2
$s3 = (0x4 * $s4) + sha3(0x0, 0x20)
$s4 = s[$s3]
$s4 = ($s4 & ((0x100 * (0 == (0x1 & $s4))) - 0x1)) / 0x2
$s5 = $m
$m = $m + (0x20 + (0x20 * ((0x1f + $s4) / 0x20)))
m[$s5] = $s4
$s6 = 0x20 + $s5
$s8 = s[$s3]
$s8 = (((0x100 * (0 == (0x1 & $s8))) - 0x1) & $s8) / 0x2
if (0 == $s8) goto 0xae0
```

```
0xfc6
-----
if (0x1f < $s8) goto 0xab5
```

```
0xab5
-----
$t = $s6
$s6 = $s6 + $s8
m[0x0] = $s3
$s7 = sha3(0x0, 0x20)
$s8 = $t
```

```
0xac3
-----
m[$s8] = s[$s7]
$t = $s7
$s7 = $s8
$s8 = 0x1 + $t
$t = $s7
$s7 = $s8
$s8 = 0x20 + $t
if ($s6 > $s8) goto 0xac3
```

```
0xfce
-----
m[$s6] = 0x100 * (s[$s3] / 0x100)
goto 0xae0
```

```
0xad7
-----
```

```
0xae0
-----

m[$m] = 0x20
$s4 = 0x20 + $m
m[$s4] = m[$s5]
$s4 = 0x20 + $s4
$t = m[$s5]
$s5 = 0x20 + $s5
$s10 = 0x0
while (0x1) {
    if ($s10 >= $t)
        break
    m[$s4 + $s10] = m[$s5 + $s10]
    $s10 = 0x20 + $s10
}
$s5 = $t
$t = $s4
$s4 = $s5
$s5 = $s5 + $t
$t = $s4
$s4 = $s5
$s5 = 0x1f & $t
if ($s5){
    $s6 = $s4 - $s5
    m[$s6] = (! ((0x100 ** (0x20 - $s5)) - 0x1)) & m[$s6]
    $s4 = 0x20 + $s6
}
return($m, $s4 - $m)
```