Memory Management

```
def f n =
  if n < 1 then
      n
  else
      (n, f (n-1))
end

f 2
```
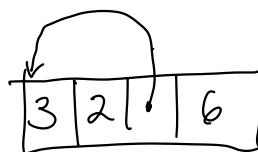
f

Allocate early

2	4	0x80069	2
2	0		

heap-cursor 0x80080

Heap

0x80050

rax 0x80051

bird main [^{n} ..

bird-main
⋮
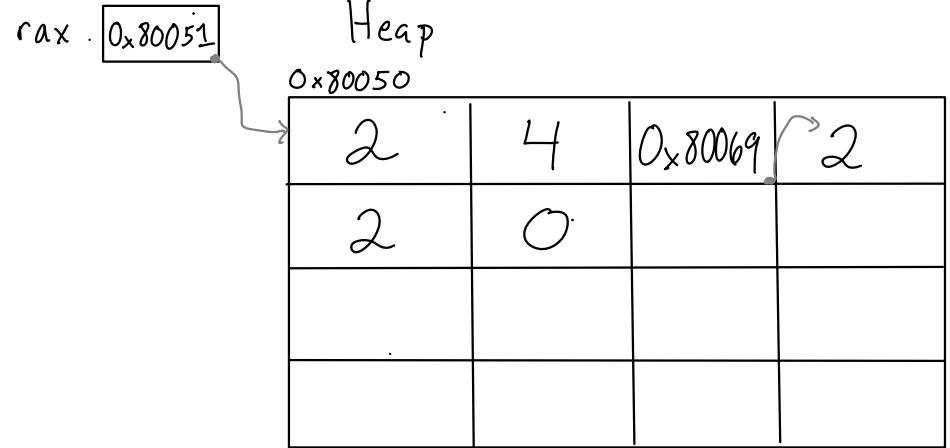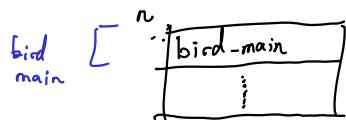
Gull extends Falcon

$\langle expr \rangle ::= \cdots$

$\qquad | \langle expr \rangle [\langle expr \rangle] := \langle expr \rangle$

$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}$
update tuple values (mutation)

let t = (1, 2, 3) in
t[1] := t

3	2		6

---

let t = (1, 2, 3) in   (1, 2, 3)
let x = t[1] in   2
let y = t[2] := 5 in   5   5

x + y + t[2]   Gull
↑   ↑   ↑
2   5   5

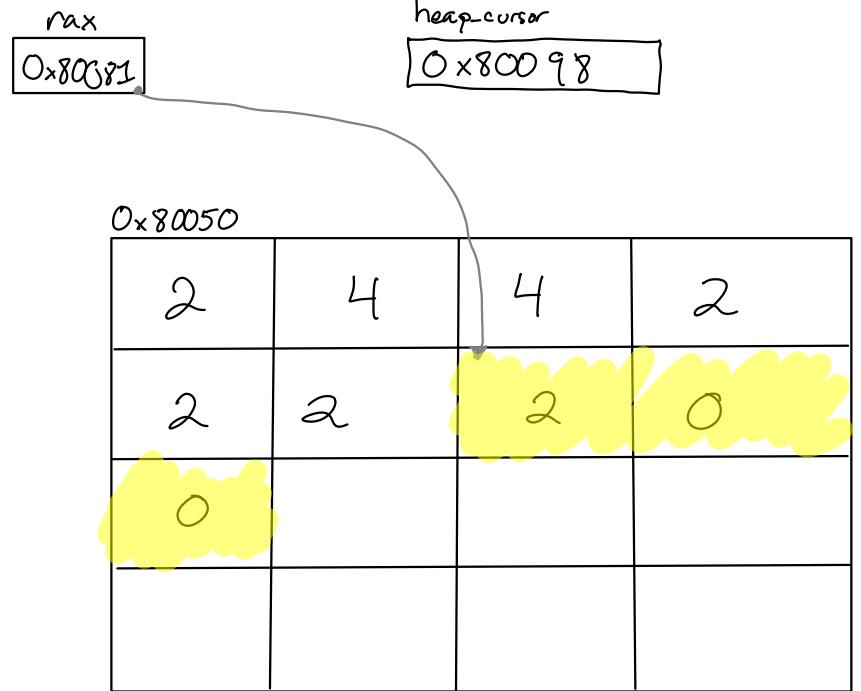
int x, y, z; $\langle expr \rangle = \langle expr \rangle$
x = (y = (z = 0))

if (x=0) { |}
   ....
}

C

Memory Management

```
def f n =
    let t = (n,n) in
    if n > 0 then
        f (n-1)
    else
        t
end

f 2
```

rax
`0x80081`

heap-cursor
`0x80098`

0x80050

2	4	4	2
2	2	2	0
0			

bird-main

How to handle waste?

1. Ask user to deallocate.
2. "Garbage collection" — figure it out ourselves