Not Eagle

Provided

```
<expr> ::= ......
         | ( <expr>, <expr> )
         | fst <expr>
         | snd <expr>
```

What other kinds of data do we want?

data structures
- lists
- tuple
- dictionary

primitives
- character (string)
- float

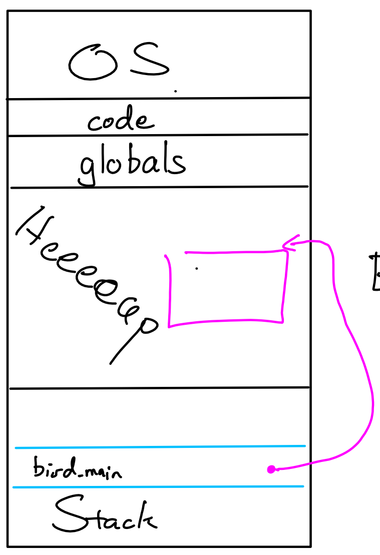
mechanics
- pointers
- references

writable vs. provided

fst (1,2) ⟹ 1
snd (1+3, 2*4) ⟹ 8
(3, true) ⟹ (3, true)
fst (1+false, 3) ⟹ ☹!!
snd (1+false, 4) ⟹ ☹!!

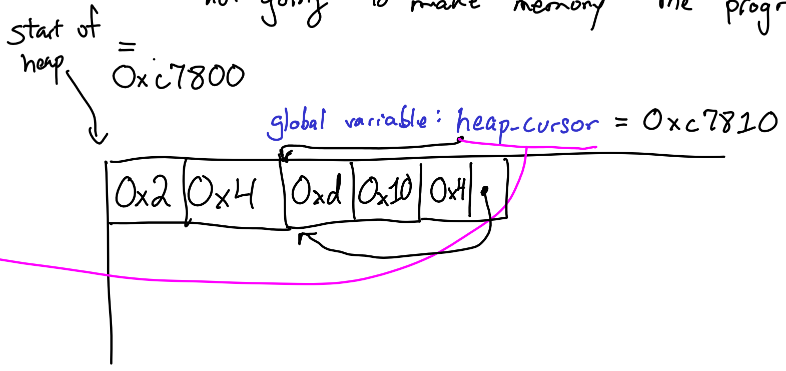
Binary Representation

Not Eagle Values	tags		Machine Values
true	0b ····11		0xFFFFFFFF FFFFFFFF
-1	0b ···· 0	-2 ~	0xFFFFFFFFFFFFFFFE
		~	0xFFFFFFFF FFFFFFF[1110]
4		8 ~	0x00 ····· 01[0000]
(1,2)	0b ····01		0xc7801
(4,(7,8))			



Eagle: malloc a bunch of memory in driver.c, pass ptr to bird_main
- allocate a bunch up front
- never allocate again
- not going to check for out-of-memory
- not going to dealloate
- not going to make memory the programmer's problem

(1,2)
(4,(7,8))



```
.section data
  align 8
  heap_cursor:
  dq 0
```

Strategy: all pointers to our heap end in 0b....000,
we mark ptrs by putting 0b...01 at end.

Base: malloc gives 8-byte aligned ptrs. Ind: all things bird are 8 bytes.