<expr> ::= ...
| isbool(<expr>)
| isint(<expr>)

Lecture representation

true = 0x8000 —— 0001
false = 0x0000 —— 0001
int n = 2n

true ↦ true
false ↦ true
5 ↦ false

isbool(~) ⟶ test bit
jz to lbl
load true
jmp end
lbl:
load false
end:

isbool(~~)

$b_{63}\ b_{62}$ —— $b_1\ b_0$

$0x80\cdots01 \mapsto 0x80\cdots01$
$0x00\cdots01 \mapsto 0x80\cdots01$
$0x00\cdots0a \mapsto 0x00\cdots01$

Notation

0x8[0010] = 0x82

in: rax
shl rax, 63
or rax, 1

$b_{63}\ b_{62}$ —— $b_1\ b_0$

b_0 0 —— 0 0

b_0 0 —— 0 1

8 1

0x[1000]0···0[0001] ↦ 0x[1000]0···0[0001]
0x[0000]0···0[0001] ↦ 0x[1000]0···0[0001]
0x00···0[1010] ↦ 0x[0000]0···0[0001]

a

Cardinal: function calls, runtime errors (Dove: fuction defns, compile errors)

Bluebird (by defn): true + true ⟹ undefined

Bluebird (reality): true + true ⟹ 1

Cardinal (ideal): true + true ⟹ !! (stop prog w/ exit code)

```
 0x80...01
+0x80...01
-----------
0x100.....02
 0x00....02
```

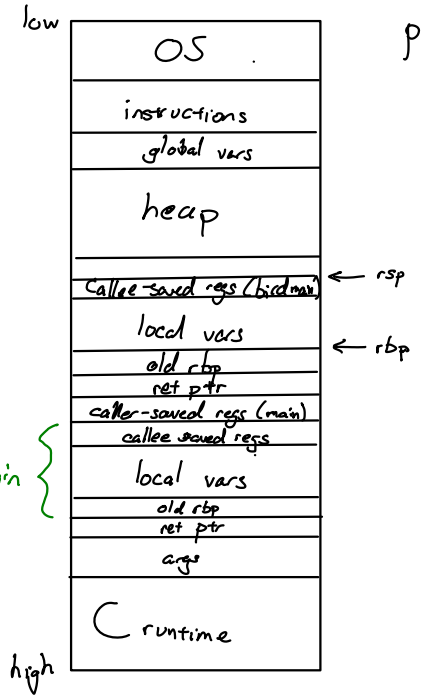
- eval left operand
  - check left operand
- store left operand
- eval right operand
  - check right operand
- perform operation

rsp = register stack pointer
rbp = register base pointer

push reg ≡ sub rsp, 8
mov [rsp], reg

pop reg ≡ mov reg, [rsp]
add rsp, 8

NOT TO SCALE



64bit POSIX C Calling Conventions
caller (main) / callee (bird_main)

caller

1a. Push caller-saved regs
- Registers are split into 2 categories
  - caller-saved / volatile regs: may change during calls

    rax, rcx, rdx, r8, r9, r10, r11
  - callee-saved / stable regs: will have same value before & after call

    rdi, rsi, rbx, rsp, rbp, r12, r13, r14, r15

1b. Assign args:
- first six: rdi, rsi, rdx, rcx, r8, r9
- rest: onto stack in rev order

1c. Ensure 16-byte alignment (skip)

1d. call

callee

2a. push rbp
2b. move rsp into rbp
2c. move rsp by amt of local mem we need
2d. save callee-saved registers
3. do your thing
4a. pop callee-saved regs
4b. move rbp into rsp
4c. pop rbp

* rax has result

caller 5. teardown stack frame

Stack mem needed?

```
mov rax, 2
mov [rbp-8], rax
mov rax, 4
mov [rbp-16], rax
mov rax, [rbp-8]
sub rax, [rbp-16]
```

16 bytes