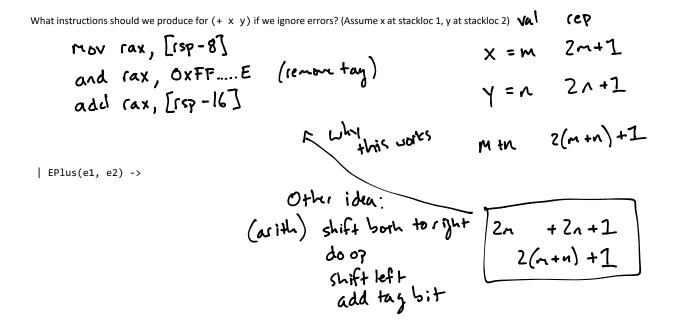
Representation (bits) Value Representation hex decimal 63-bit, 2's complement number tag bit (1=num, 0=bool) 0000 0000 0000 ... 0000 0000 0001 0011 0x000...0013 19 1111 1111 1111 ... 11<u>11</u> 1111 11**11 1101** 0xFFF...FFFD -3 32 —168 421 0000 0000 0000 ... 0000 0000 0100 0000 0x000...0045 33 true **1**111 1111 1111 ... 1111 1111 1111 111**0** 0xFFF...FFFE false **0**111 1111 1111 ... 1111 1111 1111 1110 62 bits for future value representations! true/false

What will print from main for the program that's just the number 5? For the program that's just the constant true?

type expr = | EBool of bool

I E Nun of int



```
mov (ax, [rsp-8] and rax, I cmp rax, D je op-error mov [rsp-16], (ax
```

(iden - and x with y to check both)

```
let compile (program : string) : string =
  let ast = parse program in
  let instrs = e_to_is ast 1 [] in
  let all_is = (String.concat "\n" instrs) in
  (sprintf "
section .text
global our_code_starts_here
extern print_error_and_exit
our_code_starts_here:
  %s
  ret
```

#include <stdio.h>
extern int64_t our_code_starts_here() asm("our_code_starts_here");

Void print_err_exit() {
 eprint("Error!"),
 exit(1),

" all_is)

op-error:

What if we want to get input from *outside the program*?
First mechanism – command-line arguments. How to communicate to our _code _starts _here?